WebYaST

11

www.suse.com

March 26, 2013

WebYaST Vendor Guide

WebYaST Vendor Guide

List of Authors: Jakub Friedl

Copyright © 2006–2013 Novell, Inc. and contributors. All rights reserved.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, version 1.2 or (at your option) version 1.3; with the Invariant Section being this copyright notice and license. A copy of the license version 1.2 is included in the section entitled “GNU Free Documentation License”.

For Novell trademarks, see the Novell Trademark and Service Mark list http://www.novell.com/company/legal/trademarks/tmlist.html. All other third party trademarks are the property of their respective owners. A trademark symbol (®, ™ etc.) denotes a Novell trademark; an asterisk (*) denotes a third party trademark.

All information found in this book has been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. Neither Novell, Inc., SUSE LINUX Products GmbH, the authors, nor the translators shall be held liable for possible errors or the consequences thereof.

Contents

1 Introduction 1

1.1 Customization Options 1

1.2 Supported Browsers 2

1.3 Additional Documentation and Resources 2

1.4 Feedback 2

1.5 Documentation Conventions 3

2 WebYaST Architecture 5

2.1 REST API Interface 7

2.2 Switching the REST API and Web Interface On and Off 15

2.3 SSL Certificates 16

3 Creating Appliances Containing WebYaST 19

4 Customizing WebYaST Interface 25

4.1 Replacing Graphics and Style Sheets 25

4.2 Customizing Text Strings 26

5 Configuring WebYaST Modules 29

5.1 Using Custom NTP Servers 29

5.2 Configuring System Services 29

5.3 Initial System Setup Sequence 31

5.4 Configuration of EULAs 32

5.5 Configuring Update Intervals of Status Information in Control Panel 34

5.6 Configuring the Status Module 34

5.7 Custom Link in the Control Panel 34

5.8 `/etc/webyast/vendor` Subdirectory 35

6 Additional Developer Information 37

A WebYaST Packages 39

B GNU Licenses 43

B.1 GNU Free Documentation License 43

1 Introduction

WebYaST is a Web-based remote console for controlling appliances based on SUSE® Linux Enterprise. WebYaST provides a standardized and user-friendly interface to configure the appliance “black box” environment. Whereas automation —wherever applicable—is the default procedure in the appliance environment, WebYaST is needed to perform some maintenance tasks that cannot be automated.

The set of available WebYaST modules is a subset of the YaST functionality available on SUSE® Linux Enterprise Server or SUSE® Linux Enterprise Desktop. WebYaST modules do not use the same code as the respective YaST modules and their functionality may differ. Besides, WebYaST can provide modules which are not available in YaST on SUSE Linux Enterprise Server or SUSE Linux Enterprise Desktop, for example the WebYaST Status module.

This guide is intended for vendors preparing appliances equipped with WebYaST. For information about using WebYaST to access, configure, and monitor running appliances, see the *WebYaST User Guide*.

1.1 Customization Options

The available configuration options can be limited by providing only those WebYaST modules that are necessary for the particular appliance. It is also possible to create additional custom WebYaST modules.

You can also customize the following WebYaST aspects:

- Rebrand the WebYaST interface. It is possible to replace default logos and other graphical objects.

- Change the theme of WebYaST interface with custom Cascading Style Sheets.

- Change any text strings in the WebYaST interface including their translations.

1.2 Supported Browsers

WebYaST supports the following Web browser clients:

- Mozilla Firefox, version 10.x

- Microsoft Internet Explorer, latest major release (9.x)

NOTE: Unsupported Browsers

Unsupported browsers are not blocked. If an unsupported browser is used, the user is warned by a message in his or her browser. It is recommended to update the browser in that case. The message is shown on all WebYaST pages, unless the user acknowledges and disables the message.

1.3 Additional Documentation and Resources

Many chapters in this manual contain links to additional documentation resources. These include documentation that is available on the system as well as documentation available on the Internet.

For an overview of the documentation available for your product and the latest documentation updates, refer to `http://www.novell.com/documentation`.

1.4 Feedback

Several feedback channels are available:

Bugs and Enhancement Requests

For services and support options available for your product, refer to `http://www.suse.com/support/`.

To report bugs for a product component, log in to the Novell Customer Center from `http://www.suse.com/support/` and select *My Support* > *Service Request*.

User Comments

We want to hear your comments about and suggestions for this manual and the other documentation included with this product. Use the User Comments feature at the bottom of each page in the online documentation or go to `http://www.suse.com/documentation/feedback.html` and enter your comments there.

Mail

For feedback on the documentation of this product, you can also send a mail to `doc-team@suse.de`. Make sure to include the document title, the product version and the publication date of the documentation. To report errors or suggest enhancements, provide a concise description of the problem and refer to the respective section number and page (or URL).

1.5 Documentation Conventions

The following typographical conventions are used in this manual:

- `/etc/passwd`: directory names and filenames
- *`placeholder`*: replace *`placeholder`* with the actual value
- `PATH`: the environment variable PATH
- `ls`, `--help`: commands, options, and parameters
- `user`: users or groups
- Alt, Alt + F1: a key to press or a key combination; keys are shown in uppercase as on a keyboard
- *File*, *File* > *Save As*: menu items, buttons

- *Dancing Penguins* (Chapter *Penguins*, ↑Another Manual): This is a reference to a chapter in another manual.

2 WebYaST Architecture

WebYaST consists of the following main components:

YaST2 Webservice

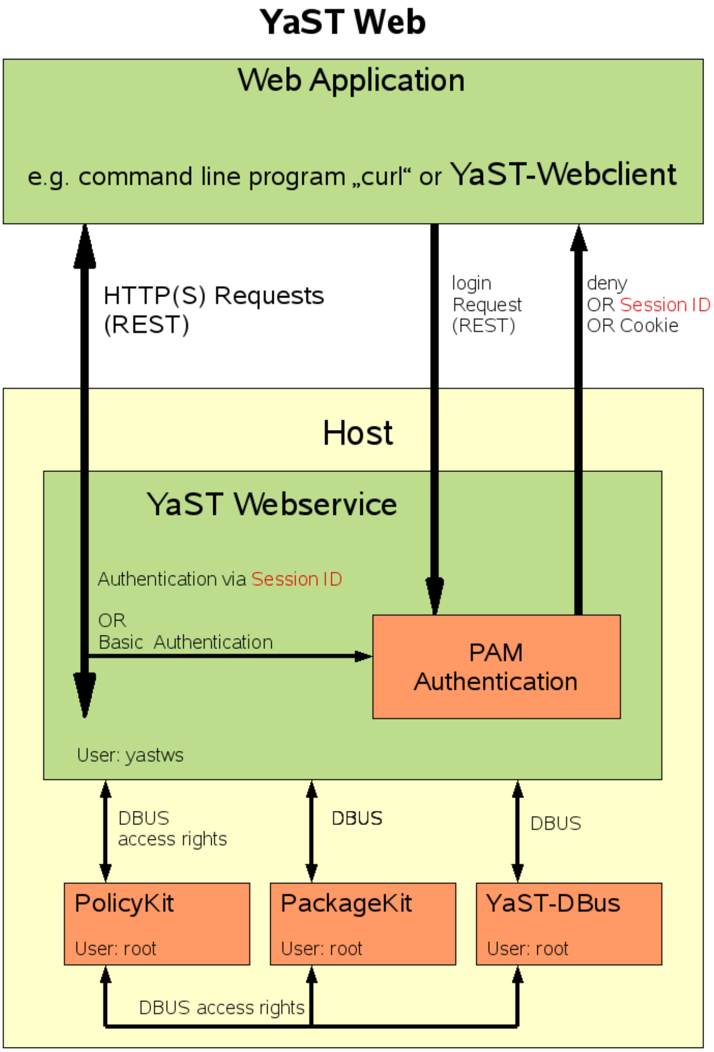
The YaST2 Webservice runs on the host system (appliance). It receives requests from YaST2 Webclient, handles user authentication via PAM modules, checks the user's rights via PolicyKit, and then sends the requests to YaST-DBus.

YaST2 Webclient

The YaST2 Webclient is a Web application that provides user interface for authentication, appliance configuration, and monitoring.

YaST2 Webclient and YaST2 Webservice communicate via the standard HTTP(S) protocol.

Figure 2.1: *WebYaST Architecture*



2.1 REST API Interface

The YaST2 Webservice provides an interface to the YaST environment of the host appliance. This service runs on an installed Web server and provides an API which is based on the REST (Representational State Transfer) architecture.

NOTE: New Web Server in Use Since WebYaST 1.2

Since version 1.2, WebYaST has used a new Web server as the back-end technology; `lighttpd`, which was used in previous releases, has been replaced with `nginx`. If you upgrade WebYaST from 1.1 to 1.2, you should remove `lighttpd` from your appliance as long as it is not needed for other applications in your appliance.

With the `nginx` Web server, WebYaST is loaded on demand and unloaded after 5 minutes of inactivity. Thus, memory is saved.

For more information about REST, see `http://en.wikipedia.org/wiki/Representational_State_Transfer`.

For example, to get or set the system time, use the following HTTP URL:

```
http://hostname/systemtime
```

The following set of operations is supported by the Web service. You can use these methods for manipulating values:

`GET`
: Retrieving a value. For more information, see Section 2.1.1.1, “The `GET` Method” (page 8).

`PUT`
: Overwriting a value. For more information, see Section 2.1.1.2, “The `PUT` Method” (page 9).

`POST`
: Creating a new value. For more information, see Section 2.1.1.3, “The `POST` Method” (page 9).

`DELETE`
: Deleting a value, for example a user account. For more information, see Section 2.1.1.4, “The `DELETE` Method” (page 10).

2.1.1 General Structure of HTTP Requests

Information can be handled in three different formats (according to the MIME type of the data supported by the Web service):

`HTML format`
: With URL `http://`*`hostname`*`/`*`systemtime`*

`XML format`
: With URL `http://`*`hostname`*`/`*`systemtime`*`.xml`

`JSON format`
: With URL `http://`*`hostname`*`/`*`systemtime`*`.json`

2.1.1.1 The `GET` Method

You can either get information in a block or you can get each entry of the block in an extra HTTP call. `GET http://`*`hostname`*`/systemtime.xml` returns:

```
<systemtime>
<currenttime type="datetime">2008-09-10T17:13:37Z</currenttime>
  <is_utc type="boolean">false</is_utc>
  <timezone>Europe/Berlin</timezone>
  <validtimezones type="array">
     <timezone>
        <id>Africa/Abidjan</id>
     </timezone>
     <timezone>
        <id>Africa/Accra</id>
     </timezone>
     ....
     ..
     .
  </validtimezones>
</systemtime>
```

GET calls can be generated on the command line as follows

```
curl -0 -X GET -b cookie_file http://0.0.0.0:4984/systemtime.xml
```

where *`cookie_file`* is a cookie containing the login information of the current session. The cookie file can be generated by the following command:

```
 curl -u tux -c cookie.txt http://0.0.0.0:4984/login
```

2.1.1.2 The `PUT` Method

PUT requests are used to overwrite existing values. Multiple values can be overwritten at once in a block or each entry of the block can be set explicitly in an extra HTTP call. For example, you can change multiple entries within the following block of XML data

```
<?xml version="1.0" encoding="UTF-8"?>
<language>
<first_language>de_DE</first_language>
<second_languages type="array">
<language>
<id>en_US</id>
</language>
<language>
<id>en_GB</id>
</language>
</second_languages>
</language>
```

by using the following command:

```
PUT http://hostname/languages
```

It is also possible to use data in HTML or JSON formats.

PUT calls can be generated with the following command:

```
curl -0 -v -H "Content-Type: application/xml; charset=utf-8"
-T data-file -b cookie_file http://0.0.0.0:4984/languages.xml
```

2.1.1.3 The `POST` Method

POST requests are used for creating new entries, for example adding a new user account. They are also used for requests that do not change any data, like starting or stopping services or generating a login session:

```
 POST http://hostname/login
```

The following call checks the system account and creates a session with given values and generates a session cookie valid for a period of 1 day:

```
<hash>
```

```
  <login>tux</login>
  <password>password_tux</password>
  <remember_me type="boolean">true</remember_me>
</hash>
```

If the login is successful, the following data are returned:

```
<hash>
  <login>granted</login>
</hash>
```

If the login is not successful, the following data are returned:

```
<hash>
  <login>denied</login>
</hash>
```

If `curl` is used for testing, you can generate a session cookie with the login data specified in *`login_xml_file`* with the following command:

```
 curl -0 -X POST -v -H "Content-Type: application/xml; charset=utf-8"
 -T login_xml_file -c cookie_file http://0.0.0.0:4984/login.xml
```

2.1.1.4 The `DELETE` Method

DELETE requests are used for deleting entries, for example a user account:

```
  DELETE http://0.0.0.0:4984/yast/v1/users/tux
```

DELETE calls can be generated with the following command:

```
  curl -0 -X DELETE -b cookie_file http://0.0.0.0:4984/users/tux
```

2.1.2 YaST2 Webservice API

The YaST2 Webservice API has two kinds of interfaces:

Fixed Interfaces
: Exist on every YaST2 Webservice. They are used for session handling or checking permissions. For more information, refer to Section 2.1.2.1, “Fixed Interfaces” (page 11).

Variable Interfaces

Depends on the particular module (user, patches, systemtime) installed on the respective YaST2 Webservice. For more information, refer to Section 2.1.2.2, “Variable Interface” (page 13).

2.1.2.1 Fixed Interfaces

The following interfaces are present in every YaST2 Webservice:

hostname/login

This interface handles login and creation of a session. An additional login cookie is generated if required. The existence of the given system user account is checked via PAM.

```
<hash>
  <login>tux</login>
  <password>password_tux</password>
  <remember_me type="boolean">true</remember_me>
</hash>
```

If the login is successful, the following data are returned:

```
<hash>
  <login>granted</login>
</hash>
```

If the login is not successful, the following data are returned:

```
<hash>
  <login>denied</login>
</hash>
```

If `curl` is used for testing, you can generate a session cookie with the login data specified in *`login_xml_file`* with the following command:

```
 curl -0 -X POST -v -H "Content-Type: application/xml; charset=utf-8"
 -T login_xml_file -c cookie_file http://0.0.0.0:4984/login.xml
```

hostname/logout

This interface destroys the current session and invalidates the respective cookie (if the cookie exists).

```
 POST hostname/logout
```

It returns:

```
<hash>
  <logout>Goodbye!</logout>
</hash>
```

hostname/permissions

This interface is useful when it is necessary to check user permissions in advance, for example, before a menu entry is generated. Two arguments are used:

`user_id`
: The ID of the user whose permissions should be checked.

`filter`
: Filter for the permissions.

The following call

```
GET /permissions?user_id=user_id
GET /permissions?user_id=user_id&filter=filter
```

returns:

```
<permissions type="array">
  <permission>
    <name>org.opensuse.yast.system.time.read</name>
    <grant type="boolean">true</grant>
  </permission>
  <permission>
    <name>org.opensuse.yast.systemtime.write-timezone</name>
    <grant type="boolean">true</grant>
  </permission>
  <permission>
    <name>org.opensuse.yast.systemtime.read-isutc</name>
    <grant type="boolean">true</grant>
  </permission>
  <permission>
    <name>org.opensuse.yast.systemtime.write</name>
    <grant type="boolean">true</grant>
  </permission>
  ...
  ..
  .
</permissions>
```

2.1.2.2 Variable Interface

WebYaST contains several plug-ins. Every plug-in is designed for a special task and has its own REST API. Because of this, the YaST2 Webservice API depends on which plug-ins (defined in packages) are installed.

A useful feature of any REST interface is that you can ask the interface for its API format via the following call:

```
 GET hostname/resources.xml
```

It returns:

```
<resources type="array">
   <resource>
     <interface>org.opensuse.yast.system.patches</interface>
     <singular type="boolean">false</singular>
     <href>/patches</href>
     </resource>

     ...

   <resource>
     <interface>org.opensuse.yast.commandlines</interface>
     <singular type="boolean">false</singular>
     <href>/yast/commandlines</href>
   </resource>
</resources>
```

2.1.3 Security Handling

The communication between YaST2 Webclient and YaST2 Webservice is secured by authentication via Pluggable Authentication Modules (PAM) and user permission checks via PolicyKit. The process flow is as follows:

1 The user starts a login request with username and password. This is sent from the user's YaST2 Webclient to the YaST2 Webservice via HTTP(S)/REST. The request is checked with PAM and the access is granted or denied accordingly. If the access is granted, a session is created and a cookie is sent to the YaST2 Webclient (on demand).

2 The authenticated user sends an HTTP(S) request with his YaST2 Webclient, for example: `GET http://hostname/yast/protocol_version/systemtime.xml`.

3 The YaST2 Webservice checks the authentication by executing the following steps:

1. Checking the `user` and `password` in the HTTP header via PAM (basic authentication).

2. Checking the `password` field for a session ID (from a former login request) which is compared with the current session.

3. If the request includes a cookie with the session information (can be generated during the login request) this cookie is compared with the current session.

If all these checks fail an HTTP error `401` is returned.

4 The YaST2 Webservice checks whether the authenticated user has the right to read these settings:

```
policyKit_check ("username",
      "read.systemtime")
```

If a call needs to be executed with `root` privileges, the request is sent using the YaST-DBus interface.

Each YaST2 Webservice call needs specific access permissions that can be defined for each user. These permissions are handled by PolicyKit in the following files: `/usr/share/PolicyKit/policy/org.opensuse.yast.*`.

If you encounter a permission error while using the API, the return code `503` is generated and the details are provided in XML:

```
<error>
        <type>NO_PERM</type>
        <description>
          Permission to allow org.opensuse.yast.system.patches.read is not
 available
          for user tux
        </description>
        <permission>org.opensuse.yast.system.patches.read</permission>
        <user>tux</user>
        <bug>false</bug>
</error>
```

where `type` contains an internal error identifier, `description` contains a human-readable description, `permission` contains the missing permission, `user` contains the affected user, and `bug` indicates if the problem is a software bug or just a problem in the configuration.

2.1.3.1 Setting Individual Permissions

Permissions can be set with the command `polkit-auth`:

```
polkit-auth --user tux --grant org.opensuse.yast.system.users.delete
```

Or they can be revoked with the following command:

```
polkit-auth --user tux --revoke org.opensuse.yast.system.users.read
```

To grant or revoke all permissions, use the script `/usr/sbin/grantwebyastrights` delivered with the package `webyast-base-ws`:

```
Usage: grantwebyastrights --user <user> --action (show|grant|revoke)
NOTE: This program should be run by user root
This call grant/revoke ALL permissions for YaST the Webservice.
In order to grant/revoke single rights use:
polkit-auth --user <user> (--grant|revoke) <policyname>
In order to show all possible permissions use:
polkit-action
```

Each YaST2 Webservice plug-in provides its own policy description file in `/usr/share/PolicyKit/policy/`. The following convention applies to the filenames: the filename is the plug-in interface name with `.policy` as suffix.

For example, the plug-in `status` has the interface name `org.opensuse.yast.system.status` (defined in `config/resources/status.yml`). The policy file in `/usr/share/PolicyKit/policy/` has the name `org.opensuse.yast.system.status.policy`.

2.2 Switching the REST API and Web Interface On and Off

By default, the REST API is turned off and the Web interface is turned on. The statuses of the REST API and Web interface are configured in the `/etc/webyast/config.yml` configuration file:

```
# Enable/disable XML REST API
# default: false
rest_api_enabled: false

# Enable/disable Web UI
# default: true
web_ui_enabled: true
```

These values can be changed in the configuration file. After any change, the service must be restarted with the `rcwebyast restart` command.

For example, to enable the REST API, follow these steps:

1. Enable REST API in the `/etc/webyast/config.yml` configuration file (`rest_api_enabled: true`).

2. Make sure that the Web interface is enabled (`web_ui_enabled: true`).

3. Restart WebYaST with the `rcwebyast restart` command.

2.3 SSL Certificates

An SSL certificate is needed to encrypt the connection between the user's browser and the YaST2 Webclient. This SSL certificate is created automatically on the first YaST2 Webservice startup (if it does not already exist).

You may want to use your own certificate or a certificate signed by your company. If the certificate files already exist, they are not changed in any way. Custom certificates can be either placed manually, or can be put into the appliance when it is created (with SUSE Studio, SUSE Studio Onsite or Kiwi).

The certificate files are located in the `/etc/lighttpd/certs/` directory. The name of the certificate file is `webyast.pem`. The name of the certificate's key file is `webyast.key`.

The file permissions should be set to `0600` for the `webyast.key` file. For the `webyast.pem` file, the permissions should be set to `0644`. The user and user group for both of these files should be set to `nginx`.

To change the certificate, you can either place your own certificate files in the `/etc/lighttpd/certs/` directory, or configure WebYaST to use a different certificate directory.

To configure a different certificate directory, edit the `ssl_certificate` and `ssl_certificate_key` options in the `/etc/webyast/nginx.conf` configuration file and restart the WebYaST server with the `rcwebyast restart` command.

3 Creating Appliances Containing WebYaST

To create appliances containing WebYaST, use either SUSE Studio or SUSE Studio Onsite®. For details, see Procedure 3.1, “Creating Appliances with SUSE Studio or SUSE Studio Onsite®” (page 21).

When creating appliances containing WebYaST, you need to include some packages for WebYaST to work. The following packages are required for the basic WebYaST framework:

- `webyast-base-ws`
- `webyast-base-ui`
- `webyast-base-ui-branding`
- `webyast-software-ws`
- `webyast-software-ui`
- `webyast-status-ws`
- `webyast-status-ui`
- `webyast-reboot-ws`
- `webyast-reboot-ui`

The following non-WebYaST package is also needed:

- syslog-ng

The following non-WebYaST package could be added for convenience:

- openssl-certs (to allow SSL connections to the NCC registration server)

NOTE: Communication with Registration Servers

When registering a system (appliance), the communication with the registration server (NCC, SMT, SLMS) is performed through encrypted HTTPS connections. The certificate that is used on the registration server needs to be trusted by the client (appliance).

This can be done by placing the CA certificate that the registration server's server certificate was signed with on the client in the /etc/ssl/certs/ directory and then running c_rehash /etc/ssl/certs/.

By default there are no certificates in /etc/ssl/certs/, as it is the administrator or security team who has to decide what certificates their appliances should trust. This means that the creator of the appliance has to take care that the certificates that his appliances should always trust are at this location. The certificates should be added as overlay files during appliance creation, and c_rehash /etc/ssl/certs/ should be run in a post creation script.

For SMT (Subscription Management Tool) and SLMS (SUSE Lifecycle Management Server) by default use the custom "YaST_CA" CA certificate that is created during installation of the server. This certificate is exposed via HTTP (but not HTTPS) at the http://*smt-or-slms-host.company.com*/smt.crt location. This file has to be put into the appliance. If a custom CA is used to sign the registration server's server certificate for SMT or SLMS, please put this CA certificate into the appliance.

For appliance registrations at NCC (Novell Customer Center) the Equifax CA certificates are needed to allow communication to the Novell registration server. The creator of the appliance could add the package openssl-certs to his appliance, which includes all needed certificates. However, this package contains many certificates and by adding this package they are all trusted as well.

The following packages are optional modules. Select them according to your needs:

- `webyast-root-user-ui`
- `webyast-licenses-ui`
- `webyast-mail-ui`
- `webyast-network-ui`
- `webyast-registration-ui`
- `webyast-services-ui`
- `webyast-time-ui`
- `webyast-root-user-ws`
- `webyast-firstboot-ws`
- `webyast-licenses-ws`
- `webyast-mail-ws`
- `webyast-network-ws`
- `webyast-ntp-ws`
- `webyast-registration-ws`
- `webyast-services-ws`
- `webyast-time-ws`

For a list of all available WebYaST packages and their functions, see Appendix A, *WebYaST Packages* (page 39).

Procedure 3.1: *Creating Appliances with SUSE Studio or SUSE Studio Onsite®*

1 In a browser, enter the URL of the publicly hosted version of SUSE Studio:`http://susestudio.com/` and log in.

Alternatively, enter the URL of your SUSE Studio Onsite server and log in.

2. As the base template for your appliance, choose *SUSE Linux Enterprise 11 > Server*.

3. Select the architecture, either *32-bit* or *64-bit*.

4. Choose a name for the new appliance and click *Create new appliance*.

5. Select the *Software* tab.

6. Click *Add repositories > Import new repository*.

7. If you are using the publicly hosted version of SUSE Studio, find the WebYaST repository in the list of the available repositories and click *Add*.

   If you are using SUSE Studio Onsite, select *Import New Repository* and enter the name and URL of your locally available mirror of the WebYaST repository. Then click *Add repository*.

   NOTE: Creating a Local Mirror of the WebYaST Repository

   The WebYaST packages are available as a pool in the Novell Customer Center. For creating a local mirror of the WebYaST repository, the Subscription Management Tool (SMT) is recommended. For more information about SMT, see the Subscription Management Tool Guide, available at `http://www.novell.com/documentation`.

8. In the *Software* tab search box, enter `webyast`. All available WebYaST packages will be listed.

   If you want to add all WebYaST packages, click *Add all*.

   To add or remove individual packages use the buttons in the left column of the list. For information about individual packages click their names in the list.

   Added packages are checkmarked.

9. Select the *Configuration* tab and configure your appliance according to your needs.

Regarding WebYaST, the default values in the following categories are mostly sufficient: *General*, *Personalize*, *Startup*, *Server*, and *Desktop*, *Storage & Memory*.

10. In the *General* section, make sure that network is enabled.

11. In the *Storage & Memory* section, set the *RAM size* at least to `512 MB`.

12. In the *Scripts* tab, activate the *Run script at the end of the build* option.

13. Enter the following script:

```
# source in profile
. /studio/profile

# enable services
insserv collectd
insserv webyast

change () {
    KEY="$1"
    VALUE="$2"
    FILE="$3"
    sed -i "s/^$KEY.*/$KEY$VALUE/" "$FILE"
}

# bnc#544694
change NETCONFIG_NTP_STATIC_SERVERS= pool.ntp.org /etc/sysconfig/
network/config
# bnc#553185
change FQDNLookup " false" /etc/collectd.conf
# bnc#553185#c1
echo "LoadPlugin df" >> /etc/collectd.conf
```

Customizing WebYaST Interface

4

You can customize the WebYaST interface by changing graphic elements such as default logos and other graphical objects, or by modifying cascading style sheets or text strings.

4.1 Replacing Graphics and Style Sheets

All graphic elements of the WebYaST interface, such as images, icons, and cascading style sheets (CSS), are stored in the YaST2 Webclient public directory:

```
/srv/www/yast/public/
```

This public directory contains the `vendor` subdirectory, which is used as an alternative location for vendor-specific images, icons, and style sheets. You can replace any file in the public directory by adding a file with the same relative path into the `vendor` subdirectory. If a file exists in this alternative location, WebYaST uses that file in place of its default equivalent.

Example 4.1: *Using Custom Graphic Files*

For example, to override the WebYaST logo stored in `/srv/www/yast/public/`*`images/webyast-logo.png`* add a replacement file in `/srv/www/yast/public/`*`vendor/images/webyast-logo.png`*.

4.2 Customizing Text Strings

WebYaST uses `gettext` to provide translations for text strings in the user interface. The translations are split in textdomains. Each textdomain contains text strings for a different part of the user interface, for example a WebYaST module.

Different textdomains are loaded for different modules. For example, when the WebYaST Status module is loaded, the following textdomains are loaded:

```
Loading standard textdomain: yast_webclient
Loading standard textdomain: yast_webclient_status
```

For each textdomain, the list of all text strings and their translations is available in a standard gettext `.po` file that you can edit. These `.po` files are available in the WebYaST source packages.

The editable `.po` files have to be converted into binary `.mo` files with the `rake makemo` command. For the main WebYaST domain, the generated `.mo` files are located in `/srv/www/yast/locale/`*`LANGUAGE`*`/LC_MESSAGES/`. For WebYaST modules, the generated `.mo` files are located in `/srv/www/yast/vendor/plugins/`*`MODULE`*`/locale/`*`LANGUAGE`*`/LC_MESSAGES/`.

This translation mechanism can also be used for customizing the text strings.

NOTE: Translations of Customized Strings

If you customize English strings, do not forget to also customize their translations, if you want to use WebYaST in languages other than English.

Procedure 4.1: *Customizing English Strings*

For example, if you want to change the English strings in the `yast_webclient` textdomain, follow these steps:

1. Copy `/srv/www/yast/po/`*`en`*`/yast_webclient.po` to `/srv/www/yast/public/vendor/text/po/`*`en`*`/yast_webclient.po`.

2. Edit the text strings in the copied file `/srv/www/yast/public/vendor/text/po/`*`en`*`/yast_webclient.po`.

3 In the /srv/www/yast/public/vendor/text directory execute the following command to create updated .mo files:

```
export RAILS_PARENT=/srv/www/yast
env LANG=en rake makemo
```

4 Restart the Web server. The following entries are generated in the log file:

```
Loading textdomain yast_webclient from public/vendor/text/locale
Loading standard textdomain yast_webclient_status
```

Configuring WebYaST Modules

5

This chapter describes how to configure the functionality of individual WebYaST modules.

The WebYaST configuration files are written in YAML language. For more information about YAML, see http://www.yaml.org.

5.1 Using Custom NTP Servers

To use a custom NTP server, enter its URL in the NETCONFIG_NTP_STATIC_SERVERS option of the /etc/sysconfig/network/config configuration file. You can add more servers, separated by commas.

5.2 Configuring System Services

The WebYaST *System Services* module manages both standard (defined in /etc/init.d) and non-standard (custom) services. The list of custom services that are managed with this module is defined in the /etc/webyast/custom_services.yml configuration file. The list of all (both standard and custom) system services that are shown in the WebYaST user interface is defined in the /etc/webyast/filter_services.yml configuration file.

5.2.1 Defining Custom Services

For each custom service, add an entry to the `/etc/webyast/custom_services.yml` file using the following format:

```
service_name: ❶
shortdescription: short_description ❷
description: description  ❸
start: "start_command" ❹
stop: "stop_command" ❺
status: "status_command" ❻
```

❶ The name of the service.
❷ An optional short description of the service (one-line summary).
❸ An optional enhanced human-readable description of the service.
❹ The command to start the service.
❺ The command to stop the service.
❻ The command to get the service status.

Example 5.1: *Example custom_services.yml File*

```
collectd:
  shortdescription: Statistics Collection Daemon
  description: Statistics Collection Daemon for filling RRD Files
  start: "/usr/sbin/rccollectd start"
  stop: "/usr/sbin/rccollectd stop"
  status: "/usr/sbin/rccollectd status"
```

NOTE: `custom_services.yml` File Permissions

Note that the `/etc/webyast/custom_services.yml` file in this example should be writable only by `root`, because start, stop and status scripts defined here are called with `root` privileges.

5.2.2 Defining Services Shown in User Interface

It is not necessary to show all available services in the WebYaST user interface. The list of services that should be shown in the WebYaST user interface can be defined in the `/etc/webyast/filter_services.yml` configuration file.

The /etc/webyast/filter_services.yml file starts with the services: line. Under this line, there is an entry for each service shown in the user interface in the following format:

```
  - service_name
```

The services are shown in the user interface in the same order as written in the file. If the /etc/webyast/filter_services.yml file is empty, all services are shown in the user interface.

Example 5.2: *Example filter_services.yml File*

```
services:
  - cron
  - my_own_service
  - nfs
  - cups
```

5.3 Initial System Setup Sequence

The initial system setup is a sequence of configuration dialogs provided by selected WebYaST modules. You can customize the sequence by removing modules from the sequence, changing the sequence order, or by adding new modules. The initial system setup sequence is defined in the configuration file /etc/webyast/basesystem.yml.

Example 5.3: *Example basesystem.yml Configuration File*

```
 steps:
  - controller: language
  - controller: eulas
  - controller: administrator
  - controller: network
  - controller: systemtime
  - controller: mail_settings
  - controller: registration
```

After the initial line, steps:, the controllers of all modules must be listed in the same order as they should appear in the initial system sequence setup. The format of each line is:

```
-controller: controller_name
```

Find the possible controller values as shipped with WebYaST in Example 5.3, “Example `basesystem.yml` Configuration File” (page 31).

It is possible to change the order or delete individual controller lines. You can also add your own controllers to the sequence.

5.3.1 Creating Custom Modules

It is possible to create your own modules for the initial system setup, using the existing initial system setup modules as a model. The modules for the initial system setup sequence must meet the following requirements:

- The form submit buttons must be implemented using the Ruby on Rails helper `form_send_buttons`. This helper generates the *Cancel*/*Save* or the *Back*/*Next* button pairs.

- After data is successfully written, the `redirect_success` method must be called.

5.4 Configuration of EULAs

With the WebYaST EULAs module, you can configure which end user license agreements (EULAs) are displayed to the user during the initial system setup.

You can configure one or more EULAs to be displayed. The list of EULAs to be used is defined in the `/etc/webyast/eulas.yml` configuration file.

Example 5.4: *Example eulas.yml File*

```
licenses:
- SLES-11
- APPLIANCE
```

After the initial line, `licenses:`, all EULAs must be listed in the same order as they appear during the initial system setup. The format of each line is:

```
- EULA_name
```

where *EULA_name* is the name of the directory containing the text of the EULA. The full path to this directory is /usr/share/webyast/eulas/licenses/*EULA_name*/.

The *EULA_name* directory must contain plain text files with the text of the respective EULA. The text files should be named according to the following pattern: license.*locale*.txt. If the file with an appropriate locale is not found, license.txt is used as a fallback.

5.4.1 Minimum EULA Configuration

Configuration with no EULAs is possible. If the /etc/webyast/eulas.yml file is missing (or if it contains only the initial line), no licenses will be displayed.

5.4.2 Minimum Single EULA Configuration

The minimum single license configuration must contain the English text of the license stored in /usr/share/webyast/eulas/licenses/*EULA_name*/license.txt and the /etc/webyast/eulas.yml configuration file with the following content:

```
licenses:
- EULA_name
```

5.4.3 Accepting EULAs

If the file /usr/share/webyast/eulas/licenses/*EULA_name*/no-acceptance-needed exists, no options for accepting or rejecting the EULA are displayed in the user interface. After clicking *Next*, the license is considered accepted.

Sometimes, accepting the EULA from command line on the target appliance is useful. Entering the following commands has the same effect as accepting the EULA via the graphical user interface:

```
touch /var/lib/webyast/eulas/accepted-licenses/EULA_name
chown webyast /var/lib/webyast/eulas/accepted-licenses/EULA_name
```

5.5 Configuring Update Intervals of Status Information in Control Panel

The system update and system health information in the Control Panel is regularly updated. The default update interval is 5 minutes. The interval can be set in the `/etc/webyast/control_panel.yml` configuration file.

The `/etc/webyast/control_panel.yml` file uses the following format (showing the default values):

```
patch_status_timeout: 28800❶
system_status_timeout: 300❷
display_patch_status: true❸
display_system_status: true❹
appliance_label: My Appliance❺
```

❶ Timeout before automatic reloading of patch status (in seconds). Value `0` disables automatic reload.

❷ Timeout before automatic reloading of system status (in seconds). Value `0` disables automatic reload.

❸ Whether the patch status should be displayed, `true` or `false`.

❹ Whether the system status should be displayed, `true` or `false`.

❺ The label shown at the top of the page (“My Appliance” by default).

5.6 Configuring the Status Module

The WebYaST Status module uses the `collectd` daemon to get the data. The daemon uses plug-ins to collect various types of status data. In the `/etc/collectd.conf` configuration file, you can configure which plug-ins are loaded.

5.7 Custom Link in the Control Panel

You can create an icon linking to an arbitrary site (such as your own Web interface) in the WebYaST control panel Configuration area.

To create such a link, create a file named `shortcuts.yml` in the top directory of any WebYaST plug-in or the YaST2 Webclient root directory. The `shortcuts.yml` file should have the following structure:

```
main:
  icon: '/icons/custom-icon.png' ❶
  url: http://example.com ❷
  groups: [ System ] ❸
  title: Title ❹
```

❶ The path to an icon file in a public directory. First, the module public directory (`/srv/www/yast/vendor/plugins/MODULE/public/`) is used. If the icon is not found there, the common WebYaST `/srv/www/yast/public/` directory is used.

❷ The URL of the link.

❸ The module group the link should belong to.

❹ The title displayed under the icon.

5.8 /etc/webyast/vendor Subdirectory

Custom vendor configuration files such as `basesystem.yml`, `filter_services.yml`, `custom_services.yml`, `eulas.yml`, `logs.yml`, or `control_panel.yml` can be placed under the `/etc/webyast/vendor` subdirectory overriding the default settings provided with the WebYaST installation. Branding packages should install modified configuration files under this directory.

6 Additional Developer Information

Additional developer information is available:

- For general information about WebYaST, see `http://en.opensuse.org/WebYaST`.
- For information about WebYaST development, see `http://en.opensuse.org/openSUSE:WebYaST_Development`.

A

WebYaST Packages

This is a complete alphabetically sorted list of RPM packages providing WebYaST services. All client-side packages depend on respective server-side packages.

Client-side Packages

`webyast-base-ui`
: YaST2 Webclient for REST-based YaST interface. No styling files are included.

`webyast-base-ui-branding`
: Styling files (CSS stylesheets and images) for WebYaST interface. A vendor can place its logo and other images and stylesheets in this packages. It replaces the default `webyast-branding-default` package.

`webyast-base-ui-branding-default`
: Default styling files (CSS stylesheets and images) for WebYaST interface.

`webyast-root-user-ui`
: User interface for setting administrator's attributes.

`webyast-licenses-ui`
: User interface for handling acceptance of EULAs.

`webyast-mail-ui`
: User interface for handling system mail settings.

`webyast-network-ui`
: User interface for setting up network.

`webyast-software-ui`
User interface for handling patches.

`webyast-roles-ui`
User interface for handling user roles (via PolicyKit).

`webyast-registration-ui`
User interface for system registration.

`webyast-firewall-ui`
User interface for handling firewall settings.

`webyast-selenium`
Selenium remote control component for testing of the user interface.

`webyast-services-ui`
User interface for handling system services.

`webyast-status-ui`
User interface for system monitoring.

`webyast-reboot-ui`
User interface for system reboot and shutdown.

`webyast-time-ui`
User interface for handling time and date settings.

`webyast-users-ui`
User interface for handling users' settings.

Server-side Packages

`webyast-base-ws`
YaST2 Webservice for REST-based YaST interface.

`webyast-root-user-ws`
REST-based interface for setting administrator's attributes.

`webyast-firstboot-ws`
REST-based interface for handling initial system settings. It is not part of the SLE11 WebYaST pattern, because it is not needed there (the initial configuration is typically performed during system installation).

`webyast-licenses-ws`
REST-based interface for handling acceptance of EULAs.

`webyast-mail-ws`
REST-based interface for handling mail settings.

`webyast-network-ws`
REST-based interface for handling network settings.

`webyast-ntp-ws`
REST-based interface for basic NTP access.

`webyast-software-ws`
REST-based interface for handling installation of patches and packages.

`webyast-registration-ws`
REST-based interface for system registration at Novell Customer Center, Subscription Management Tool or SUSE Lifecycle Management Server.

`webyast-firewall-ws`
REST-based interface for handling firewall settings.

`webyast-services-ws`
REST-based interface for handling system services.

`webyast-status-ws`
REST-based interface for system monitoring.

`webyast-reboot-ws`
REST-based interface for basic system access, restarting and shutting down.

`webyast-time-ws`
REST-based interface for handling system time and date.

`webyast-users-ws`
REST-based interface for handling users' settings.

B

GNU Licenses

This appendix contains the GNU General Public License version 2 and the GNU Free Documentation License version 1.2.

GNU Free Documentation License

Copyright (C) 2000, 2001, 2002 Free Software Foundation, Inc. 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA. Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. 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. Secondarily, 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.

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.

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 3.

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

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.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 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.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 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. 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.

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 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.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 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 4) to Preserve its Title (section 1) will typically require changing the actual title.

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.

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.

ADDENDUM: How to use this License for your documents

```
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.