

Guardian Digital WebTool API Guide

Guardian Digital WebTool API Guide

Revision History

Revision \$Revision: 1.5 \$ Revised by: AP, RWM

Table of Contents

1. Introduction	1
2. WebTool Core API Reference	3
2.1. WebTool	3
2.1.1. NAME	3
2.1.2. SYNOPSIS	3
2.1.3. DESCRIPTION	3
2.1.4. METHODS	3
2.1.5. AUTHORS	5
2.1.6. COPYRIGHT AND LICENSE	5
2.2. WebTool::UI	5
2.2.1. NAME	5
2.2.2. SYNOPSIS	5
2.2.3. DESCRIPTION	5
2.2.4. METHODS	5
2.2.5. AUTHORS	13
2.2.6. COPYRIGHT AND LICENSE	13
2.3. WebTool::UI::Std	13
2.3.1. NAME	13
2.3.2. SYNOPSIS	13
2.3.3. DESCRIPTION	14
2.3.4. METHODS	14
2.3.5. AUTHORS	15
2.3.6. COPYRIGHT AND LICENSE	15
2.4. WebTool::UI::Popup	15
2.4.1. NAME	15
2.4.2. SYNOPSIS	15
2.4.3. DESCRIPTION	15
2.4.4. METHODS	15
2.4.5. AUTHORS	16
2.4.6. COPYRIGHT AND LICENSE	16
2.5. WebTool::ACL	16
2.5.1. NAME	16
2.5.2. SYNOPSIS	16
2.5.3. DESCRIPTION	16
2.5.4. METHODS	16
2.5.5. AUTHORS	18
2.5.6. COPYRIGHT AND LICENSE	18
2.6. WebTool::Constants	18
2.6.1. NAME	18
2.6.2. SYNOPSIS	18
2.6.3. DESCRIPTION	19
2.6.4. AUTHORS	19
2.6.5. COPYRIGHT AND LICENSE	19
2.7. WebTool::MySQL	19
2.7.1. NAME	19
2.7.2. SYNOPSIS	19
2.7.3. DESCRIPTION	19
2.7.4. METHODS	19

2.7.5. AUTHORS.....	20
2.7.6. COPYRIGHT AND LICENSE.....	20
3. WebTool Module API Reference.....	21
3.1. apache.....	21
3.1.1. NAME.....	21
3.1.2. SYNOPSIS.....	21
3.1.3. DESCRIPTION.....	21
3.1.4. METHODS.....	21
3.1.5. AUTHORS.....	26
3.1.6. COPYRIGHT AND LICENSE.....	26
3.2. datetime.....	26
3.2.1. NAME.....	26
3.2.2. SYNOPSIS.....	26
3.2.3. DESCRIPTION.....	26
3.2.4. METHODS.....	26
3.2.5. AUTHORS.....	28
3.2.6. COPYRIGHT AND LICENSE.....	28
3.3. firewall.....	28
3.3.1. NAME.....	28
3.3.2. SYNOPSIS.....	28
3.3.3. DESCRIPTION.....	28
3.3.4. METHODS.....	29
3.3.5. AUTHORS.....	35
3.3.6. COPYRIGHT AND LICENSE.....	35
3.4. ftp.....	36
3.4.1. NAME.....	36
3.4.2. SYNOPSIS.....	36
3.4.3. DESCRIPTION.....	36
3.4.4. METHODS.....	36
3.4.5. AUTHORS.....	37
3.4.6. COPYRIGHT AND LICENSE.....	38
3.5. libwrap.....	38
3.5.1. NAME.....	38
3.5.2. SYNOPSIS.....	38
3.5.3. DESCRIPTION.....	38
3.5.4. METHODS.....	38
3.5.5. AUTHORS.....	39
3.5.6. COPYRIGHT AND LICENSE.....	39
3.6. logging.....	39
3.6.1. NAME.....	39
3.6.2. SYNOPSIS.....	39
3.6.3. DESCRIPTION.....	40
3.6.4. METHODS.....	40
3.6.5. AUTHORS.....	41
3.6.6. COPYRIGHT AND LICENSE.....	41
3.7. named.....	41
3.7.1. NAME.....	41
3.7.2. SYNOPSIS.....	41
3.7.3. DESCRIPTION.....	41

3.7.4. METHODS	41
3.7.5. AUTHORS	46
3.7.6. COPYRIGHT AND LICENSE	47
3.8. network	47
3.8.1. NAME	47
3.8.2. SYNOPSIS	47
3.8.3. DESCRIPTION	47
3.8.4. METHODS	47
3.8.5. AUTHORS	52
3.8.6. COPYRIGHT AND LICENSE	52
3.9. postfix	52
3.9.1. NAME	52
3.9.2. SYNOPSIS	52
3.9.3. DESCRIPTION	53
3.9.4. METHODS	53
3.9.5. AUTHORS	59
3.9.6. COPYRIGHT AND LICENSE	59
3.10. reporting	59
3.10.1. NAME	59
3.10.2. SYNOPSIS	59
3.10.3. DESCRIPTION	59
3.10.4. METHODS	60
3.10.5. AUTHORS	60
3.10.6. COPYRIGHT AND LICENSE	60
3.11. services	60
3.11.1. NAME	61
3.11.2. SYNOPSIS	61
3.11.3. DESCRIPTION	61
3.11.4. METHODS	61
3.11.5. AUTHORS	64
3.11.6. COPYRIGHT AND LICENSE	64
3.12. ssh	64
3.12.1. NAME	64
3.12.2. SYNOPSIS	64
3.12.3. DESCRIPTION	64
3.12.4. METHODS	64
3.12.5. AUTHORS	65
3.12.6. COPYRIGHT AND LICENSE	65
3.13. sysstat	66
3.13.1. NAME	66
3.13.2. SYNOPSIS	66
3.13.3. DESCRIPTION	66
3.13.4. METHODS	66
3.13.5. AUTHORS	71
3.13.6. COPYRIGHT AND LICENSE	71
3.14. users	71
3.14.1. NAME	71
3.14.2. SYNOPSIS	71
3.14.3. DESCRIPTION	71
3.14.4. METHODS	71

3.14.5. AUTHORS.....	74
3.14.6. COPYRIGHT AND LICENSE.....	74
3.15. webtool.....	74
3.15.1. NAME.....	74
3.15.2. SYNOPSIS.....	74
3.15.3. DESCRIPTION.....	74
3.15.4. METHODS.....	75
3.15.5. AUTHORS.....	77
3.15.6. COPYRIGHT AND LICENSE.....	77

Chapter 1. Introduction

The Guardian Digital WebTool is a web-based administration toolkit for EnGarde Secure Linux. It is carefully engineered to configure and maintain the secure services provided by EnGarde Secure Linux and to guide even the most inexperienced system administrator in performing common system tasks -- from simply adding a user all the way up to setting up and configuring a secure Web site.

The WebTool also has a very comprehensive Perl API (Application Programming Interface) which allows for rapid development of feature-rich modules in many different languages. The rest of this document is a reference to this interface and is broken down into two main sections:

- **WebTool Core API Reference**

This sections documents the `WebTool::*` namespace. These Perl modules are used for constructing pages and for general I/O-related tasks such as reading and writing files.

- **WebTool Module API Reference**

This sections documents the interface for each existing WebTool module. Each module has it's own private namespace with methods specific to it -- for example, the *named* module has methods for DNS zone maintenance and the *libwrap* module has methods for manipulating `/etc/hosts.allow`.

This document is intended for people who are interested in extending the Guardian Digital WebTool by writing new interfaces and for people who are interested in using the API for other applications. If you are interested in contributing code to Guardian Digital, please contact us!

Now, on with the show...

Chapter 2. WebTool Core API Reference

This section documents the core WebTool::* namespace. The main module, WebTool, contains general methods which are applicable across all of the modules (read_file, write_file, audit,etc.)

The WebTool::UI namespace contains methods for creating, translating, and drawing pages to the end-user.

Finally, the WebTool::ACL module provides an interface to the Guardian Digital WebTool Access Control subsystem, the WebTool::Constants module provides, well, constants, and the WebTool::MySQL module provides an interface to the MySQL daemon.

2.1. WebTool

2.1.1. NAME

WebTool - The core WebTool API.

2.1.2. SYNOPSIS

```
use WebTool;
```

2.1.3. DESCRIPTION

This module provides the core WebTool API. Functions for loading WebTool modules, creating temporary files and directories, auditing WebTool operations, and reading/writing files are all provided via a non-Object Oriented interface.

The audit(\$) interface is exported by default.

2.1.4. METHODS

audit(\$)

Basic function is to append WEBTOOL_AUDIT_LOG file. It receives one optional argument (message). In file log is written in order of Timestamp,user,filename,line.

Example usage:

```
$service = 'named';  
if ($services->start($service) == 0) {  
    audit("Successfully started the '$service' service.");  
}
```

Entry in webtool-audit.log will be something like this:

```
[2006/07/17 09:33:02] admin modules/services/toggle_current.cgi:42  
Successfully started the 'named' service.
```

load_module(\$)

Basic function is to load particular module. It receives one argument (module). New module is loaded into its own namespace. This function must be called from within BEGIN{ } block. In a result module is actually loaded using "use \$module" as return value.

Example usage:

```
BEGIN { WebTool::load_module('services');
```

mktemp(\$)

This function makes temporary filename. It takes a single, optional argument on what to use as the base filename(template). If you don't specify this base then 'webtool' will be used. It returns a result from running a mktemp command under WEBTOOL_TMPDIR/argument.XXXXXX .

mkdtemp(\$)

This function generates uniquely named temporary directory. It takes a single, optional argument on what to use as the base filename(template). If you don't specify this base then 'webtool' will be used. It returns a result from running a mkdtemp command under WEBTOOL_TMPDIR/argument.XXXXXX .

Example usage:

```
my $tmpfile = WebTool::mktemp('apt-get');  
my $ajax    = new CGI::Ajax('progress' => "progress.cgi?tmpfile=$tmpfile");
```

output_ctl(\$)

One argument is passed(0 or 1) to this function, according to that value function open/close STDOUT/STDERR and flushes.

Example usage:

```
output_ctl(0);  
output_ctl(1);
```

read_file(\$)

This function receives one argument(file). It opens the file and read it line by line and puts every line in array. It returns array.

Example usage:

```
my @file = WebTool::read_file($file);
```

write_file(\$)

This function receives two argument(file and array). It opens the file and writes in line by line from array.

Example usage:

```
WebTool::write_file(UPS_CONF, @f);
```

2.1.5. AUTHORS

Ryan W. Maple <ryan@guardiandigital.com>

2.1.6. COPYRIGHT AND LICENSE

Copyright Guardian Digital, Inc., All Rights Reserved

2.2. WebTool::UI**2.2.1. NAME**

WebTool::UI - The main Guardian Digital WebTool user interface class.

2.2.2. SYNOPSIS

```
use WebTool::UI;
```

2.2.3. DESCRIPTION

This module contains the core WebTool user-interface API and is not normally called directly. WebTool::UI::Std and WebTool::UI::Popup inherit from this class, therefore all of the functions present in this module are available in them.

2.2.4. METHODS**disable_construct_page()**

This function disables current object's construct page property.

Example usage:

```
my $page->disable_construct_page();
```

enable_construct_page()

This function enables current object's construct page property.

Example usage:

```
my $page->enable_construct_page();
```

disable_http_header()

This function disables current object's 'show http header' property.

Example usage:

```
my $page->disable_http_header();
```

disable_page_top()

This function disables current object's 'show_top' property. Its usually called to hide the top layout of page.

Example usage:

```
my $page->disable_page_top();
```

disable_page_bottom()

This function disables current object's 'show_bottom' property. Its usually called to hide the bottom layout of page.

Example usage:

```
my $page->disable_page_bottom();
```

set_page_title(\$)

This function sets current page's 'title' property. If this optional argument is not passed than default page title from global constant DEFAULT_PAGE_TITLE is set.

Example usage:

```
my $title = "Guardian Digital Inc.";
my $page->set_page_title($title);
```

set_page_top(\$)

This function sets current page's top layout to data provided as argument.

Example usage:

```
my $data = "Guardian Digital Inc.";
my $page->set_page_top($data);
```

set_page_bottom(\$)

This function sets current page's bottom layout to data provided as argument.

Example usage:

```
my $data = "Guardian Digital Inc.";
my $page->set_page_bottom($data);
```

hide_selects()

This function enables javascript code which will hide SELECT objects when the navigation bar is active.

Example usage:

```
$page->hide_selects();
```

unhide_selects()

This function disables javascript code which will unhide SELECT objects when the navigation bar is inactive.

Example usage:

```
$page->unhide_selects();
```

enable_overlib()

This function is used to enable the javascript code which shows purpol box "over" current page showing some extra information.

Example usage:

```
$page->enable_overlib();
```

disable_overlib()

This function is used to disable the javascript code which shows purpol box "over" current page showing some extra information.

Example usage:

```
$page->disable_overlib();
```

tmpl_load(\$)

This function receives html template file as argument and loads it.

Example usage:

```
$page->tmpl_load($tmpl_file);
```

tmpl_set(\$\$)

This function receives key and value as arguments and sets html template's variables accordingly.

Example usage:

```
$page->tmpl_set('title' => 'Edit Local Group');
```

tmpl_translate(\$)

This function takes a string as an argument and returns the translated form of it as per the lang/<language> file.

Example usage:

```
$title = $page->tmpl_translate('viewstatustitle');
```

send_http_header(\$)

This function is basically used for setting desired http header(content type). This function receives one optional argument header type. Default value is 'text/html'.

Example usage:

```
$page->send_http_header();
```

draw_page()

According to the properties set, This function translates and return the page if only asked to construct_page, otherwise actually print it to the browser.

Example usage:

```
$page->draw_page();
```

err(\$@@)

This function is used to display an error pop-up using JavaScript. The first argument is the error message that you want to appear in the pop-up. The other two option arguments are whether or not it's fatal (default is true) and how far back the JavaScript should take them (default is -1, or back one page).

Example usage:

```
my $msg = "An error Occured";
$page->err($msg);
```

close_popup()

This function is used to close the popup using javascript.

Example usage:

```
$page->close_popup();
```

popup_button(\$\$@@@)

This function creates HTML BUTTON(SUBMIT) element so that pop-up window can be opened when its clicked. It receives five arguments(2 mandetory and 3 optional). From those link and display(value property of button) are mandetory whereas width, height and key are optional.

Example usage:

```
my $create = $page->popup_button('edit_route.cgi', 'create_route', 550, 250);
```

popup_link(\$\$@@@)

This function creates HTML HYPERLINK element so that pop-up window can be opened when its clicked. It receives five arguments(2 mandetory and 3 optional). From them link and display(text to be displayed) are mandetory whereas width, height and key are optional.

Example usage:

```
$page->popup_link("edit_client.cgi", "edit client", 400, 300);
```

make_button_link(\$\$)

This function creates a HTML BUTTON(SUBMIT) element which function like a link so that it opens a target location when clicked. It receives target location and display string as arguments.

Example usage:

```
my $button = $page->make_button_link('edit_interface.cgi','CLICK ME');
```

make_overlib(\$@)

This function reformats it's argument into a format suitable for being stuffed into overlib. Typically this involves escaping any javaScript metacharacters (', ") and converting any newlines (\r\n, \n) to
eaks. It then returns JavaScript code which is suitable for displaying the overlib.It receives two arguments (content and width), Default value for width is 200.

Example usage:

```
my $rv = $page->make_overlib($table, $size);
```

make_link(\$\$)

This function simply creates a HTML HYPERLINK ELEMENT. It receives two arguments (target and display text).

Example usage:

```
$page->make_link("edit_slave_zone.cgi", $name);
```

input_button(\$\$)

This function creates a HTML BUTTON element. It receives two button properties as arguments (name and value).

Example usage:

```
$page->input_button('create', 'Create Host');
```

input_checkbox(\$\$\$)

This function creates HTML CHECKBOX element. It receives three checkbox properties as arguments (name, value, checked/unchecked).

Example usage:

```
$page->input_checkbox('shutdown_confirm', 1, 0);
```

input_hidden(\$@)

This function creates HTML HIDDEN element. It receives two properties as arguments (name, value).

Example usage:

```
$page->input_hidden('device_orig', $route->{'device'});
```

input_password(\$\$@@)

This function creates HTML PASSWORD field. It receives four property values as arguments (name, value, size and maximum length).

Example usage:

```
$page->input_password('password1', "", 15);
```


input_radio(\$\$\$)

This function creates HTML RADIOBUTTON element. It receives three arguments (name, value, checked/unchecked).

Example usage:

```
$page->input_radio('usessl', 'yes', 'no');
```

input_select(\$\$\$)

This function creates HTML SELECT element. It receives three arguments. Name, default selected value and array reference of all the selection option.

Example usage:

```
$page->input_select('language', $selected, \@languages);
```

input_text(\$\$\$@)

This function creates HTML TEXT element. It receives four arguments: name,value,size,maxlength.

Example usage:

```
$page->input_text('username', $username, 15, 20);
```

input_textarea(\$\$@@@)

This function creates HTML TEXTAREA element. It receives five arguments: name,value,rows(default 6), columns(default 30) and wrap(default 'virtual').

Example usage:

```
$page->input_textarea('allowfrom', '127.0.0.1/32', 4, 14);
```

input_upload(\$)

This function creates HTML INPUTBOX of type FILE. It receives name as an argument.

Example usage:

```
$page->input_upload('key_up');
```

input_icb(\$@@@)

This function creates an interface chooser box. Basically it returns a button that we show on page, clicking on this button will show a pop up of all the interfaces currently defined on the system. From pop up we can select an interface and selection will be placed in input box of calling page.

Example usage:

```
$page->input_icb('iface', 'address', 0));
```

input_gcb(\$@@)

This function creates an group chooser box. Basically it returns a button that we show on page, clicking on this button will show a pop up of all the groups the system. From pop up we can select a group and selection will be placed in input box of calling page.

Example usage:

```
$page->input_gcb('group', 0);
```

input_ucb(\$@@)

This function creates an user chooser box. Basically it returns a button that we show on page, clicking on this button will show a pop up of all the users the system. From pop up we can select a user and selection will be placed in input box of calling page.

Example usage:

```
$page->input_ucb('recipient', 0, $cnt);
```

make_textarea_array(\$)

This function receives data from textarea, removes newlines, splits data on white spaces and gives back data in array form.

Example usage:

```
my @dns_search = $page->make_textarea_array($textarea);
```

process_form()

This function returns hash containing form element/data as key/value pairs.

Example usage:

```
my %in = WebTool::UI::process_form();
```

process_form_mime()

This function works the same as process_form but it handles multipart/form-data encoded data, typically file uploads. It parses form data and returns a hash containing key-value pairs.

Example usage:

```
my %in = WebTool::UI::process_form_mime();
```

redirect(\$)

This function redirects to page/url provided as argument.

Example usage:

```
WebTool::UI::redirect('list_groups.cgi');
```

str_truncate(\$@)

This function truncates the string on the basis of parameters passed and returns the truncated string. default truncation length is 20.

Example usage:

```
my $gcos = $page->str_truncate($string, 20);
```

kill_whitespace(\$)

This function removes the white spaces from passed argument string.

Example usage:

```
my $s_comment = $page->kill_whitespace("Remove White Space From Me");
```

2.2.5. AUTHORS

Ryan W. Maple <ryan@guardiandigital.com>

2.2.6. COPYRIGHT AND LICENSE

Copyright Guardian Digital, Inc., All Rights Reserved

2.3. WebTool::UI::Std

2.3.1. NAME

WebTool::UI::Std - Guardian Digital WebTool user interface class for standard windows/pages.

2.3.2. SYNOPSIS

```
use WebTool::UI::Std;  
my $page = new WebTool::UI::Std;
```

2.3.3. DESCRIPTION

This module contains UI methods for handling standard windows/pages in the Guardian Digital WebTool. This module inherits from WebTool::UI and overloads/customizes keys API calls specific to standard windows.

This is the module that most of your code will use.

2.3.4. METHODS

draw()

This function basically makes a page by creating page top and bottoms.

Example usage:

```
my $page->draw();
```

make_page_top()

This function makes top layout of the page using html template(PAGE_TOP_TMPL) and sets template variables related to title, navigation, overlib, backlink.

Example usage:

```
my $page->make_page_top();
```

make_page_bottom()

This function makes bottom layout of the page using html template(PAGE_BOTTOM_TMPL).

Example usage:

```
my $page->make_page_bottom();
```

make_page_menu()

If Context menu is registered, then this function makes main menu of the webtool which contains services, system, wizards, auditing etc.

Example usage:

```
my $self->make_page_menu();
```

register_backlink(\$)

This function registers/sets backlink. It receives one optional argument(link) If argument is given than it sets that argument as backlink else it registers '/modules/index/' as backlink.

Example usage:

```
$page->register_backlink('index.cgi');
```

2.3.5. AUTHORS

Ryan W. Maple <ryan@guardiandigital.com>

2.3.6. COPYRIGHT AND LICENSE

Copyright Guardian Digital, Inc., All Rights Reserved

2.4. WebTool::UI::Popup**2.4.1. NAME**

WebTool::UI::Popup - Guardian Digital WebTool user interface class for "popup" windows.

2.4.2. SYNOPSIS

```
use WebTool::UI::Popup;
my $page = new WebTool::UI::Popup;
```

2.4.3. DESCRIPTION

This module contains UI methods for handling "popup" windows in the Guardian Digital WebTool. This module inherits from WebTool::UI and overloads/customizes keys API calls specific to popups.

2.4.4. METHODS**draw()**

This function generates a page by making its top and bottom layout.

Example Usage:

```
$page->draw();
```

make_page_top()

This function makes a standard TOP portion of the page using HTML template(PAGE_TOP_TMPL) and sets the page title.

Example Usage:

```
$page->make_page_top();
```

make_page_bottom()

This function makes a standard BOTTOM portion of the page using HTML template(PAGE_BOTTOM_TMPL).

Example Usage:

```
$page->make_page_bottom();
```

2.4.5. AUTHORS

Ryan W. Maple <ryan@guardiandigital.com>

2.4.6. COPYRIGHT AND LICENSE

Copyright Guardian Digital, Inc., All Rights Reserved

2.5. WebTool::ACL

2.5.1. NAME

WebTool::ACL - Access control subsystem for the Guardian Digital Webtool.

2.5.2. SYNOPSIS

```
use WebTool::ACL;

my $acl      = new WebTool::ACL;
my $rv      = $acl->check_fatal();
```

2.5.3. DESCRIPTION

This module contains functions for controlling access to parts of the Guardian Digital WebTool. End-user code should never have to call these functions directly -- WebTool::ACL is used internally by the WebTool::UI module when a page is loaded.

Each file in the WebTool is assigned a major/minor number and an access right. If a user does not have the correct privileges to access the given file (when WebTool::UI pulls up the page), an error page is shown.

2.5.4. METHODS

check_fatal()

This function basically checks access rights for the combination of user and file. Function obtains these values from environment. It returns true or false.

Example Usage:

```
if (check_fatal()) {
    ....
}
```

err(\$)

This function receives a custom error message as argument. It returns javascript pop up containing error message with button(BACK button).

Example Usage:

```
$acl->err($msg);
```

fetch_module_acl(\$\$)

This function receives a module and script as arguments. Using that module name it reads that module's acl file, and gets/returns a hash reference of values of major,minor,access associated with file.

Example Usage:

```
my $acl = $acl->fetch_module_acl($module, $file);
```

process_environment()

This function returns current user, current script and current module from environment(%ENV).

Example Usage:

```
my ($user, $module, $file) = $self->process_environment();
```

read_module_acl_file(\$)

This function takes module name as argument. Basically it reads ACL file of particular module. Hash reference is created associating each file with major,minor and access values and returned.

Example Usage:

```
my $tst = $acl->read_module_acl_file($module);
$major_value = $tst->{$file}->{'major'};
```

```
$minor_value = $tst->{$file}->{'minor'};  
$access_value = $tst->{$file}->{'access'};
```

acl_add(\$\$\$\$)

This is ACL manipulation method. Basically adding access control list for user. This function takes username,major,minor,access as argument. Value of access argument should be 1 or 2 (1=read only, 2=read/write).

Example Usage:

```
$acl->acl_add($username, $major, $minor, $access);
```

acl_clear(\$)

This function is used to delete the access control list of particular user. This function takes username as argument.

Example Usage:

```
$acl->acl_clear($username);
```

acl_get(\$)

This function is used to get access control list(as hash reference) of particular user. This function takes username as argument. In hash key is formed as \$major:\$minor and \$access is assigned as respective value.

Example Usage:

```
$acl = $acl->acl_get($username);
```

2.5.5. AUTHORS

Ryan W. Maple <ryan@guardiandigital.com>

2.5.6. COPYRIGHT AND LICENSE

Copyright Guardian Digital, Inc., All Rights Reserved

2.6. WebTool::Constants

2.6.1. NAME

WebTool::Constants - Guardian Digital WebTool constants.

2.6.2. SYNOPSIS

```
use WebTool::Constants;

if (-f WEBTOOL_UNCONFIGURED) {
    WebTool::UI::redirect('/modules/initconfig/');
    exit 0;
}
```

2.6.3. DESCRIPTION

This module provides various constants which are used by the rest of the Guardian Digital WebTool. Some constants are exported by default while some need to be accessed via the `WebTool::Constant::*` Namespace.

2.6.4. AUTHORS

Ryan W. Maple <ryan@guardiandigital.com>

2.6.5. COPYRIGHT AND LICENSE

Copyright Guardian Digital, Inc., All Rights Reserved

2.7. WebTool::MySQL

2.7.1. NAME

WebTool::MySQL - Methods for interfacing with the MySQL daemon.

2.7.2. SYNOPSIS

```
my $mysql = new WebTool::MySQL;

if ($mysql->is_installed() && defined($password)) {
    $mysql->change_root_password($password);
}
```

2.7.3. DESCRIPTION

This module contains functions for interfacing with the running MySQL daemon. The WebTool talks to the MySQL daemon via the special 'engarde-maint' MySQL user which has full privileges.

2.7.4. METHODS

is_installed()

Function checks to see if MySQL(checks daemon mysqld) is installed or not, and returns relevant result(true/false).

Example Usage:

```
if ($mysql->is_installed()) {  
    ....  
}
```

change_root_password(\$)

This function is used to change the password of root in MySQL. It receives a new password as an argument. Changes take effect immediately.

Example usage:

```
$mysql->change_root_password($new_password);
```

mysql_connect()

This subroutine is used to connect to MySQL database. It returns database handle as result.

Example Usage:

```
my $dbh = $mysql->mysql_connect();
```

mysql_disconnect(\$)

This subroutine is used to disconnect MySQL DB, it requires DB handle to be passed as argument.

Example Usage:

```
$mysql->mysql_disconnect($dbh);
```

2.7.5. AUTHORS

Ryan W. Maple <ryan@guardiandigital.com>

2.7.6. COPYRIGHT AND LICENSE

Copyright Guardian Digital, Inc., All Rights Reserved

Chapter 3. WebTool Module API Reference

This section documents the interface for each existing WebTool module. Note that each "WebTool module" in `/usr/webtool/modules` has it's own associated Perl module. This allows any given module (ie, *apache*) to import another module (ie, *services*) and use it's functionality (ie, starting and stopping the httpd service).

3.1. apache

3.1.1. NAME

apache - Guardian Digital WebTool module interface to the Apache webserver.

3.1.2. SYNOPSIS

```
use apache;
my $apache = new apache;
```

3.1.3. DESCRIPTION

This Guardian Digital WebTool module provides an interface to manipulating the Apache webserver. This module allows you to manage virtual hosts, databases for virtual hosts, and SSL certificates for SSL virtual hosts.

3.1.4. METHODS

is_installed()

This function is used to check that apache web server is installed or not. It returns true(1) or false(0).

Example Usage:

```
if ($apache->is_installed()) .....
```

mysql_installed()

This function is used to check that MySQL server is installed or not. It returns true(1) or false(0).

Example Usage:

```
if ($apache->mysql_installed()) ...
```

get_vhosts()

This function reads apache configuration file and returns information about all NameVirtualhosts and Virtualhosts(address, port, alias,redirect etc) as key-value pair in hash reference.

Example Usage:

```
my $vhosts = $apache->get_vhosts();
```

create_vhost(\$\$)

This function basically creates a virtual host(makes virtual host entry to httpd.conf). It receives two hash references, one contains information like hostname, port, webmaster, group etc information. Another hash reference argument contains existing virtual hosts information(from get_vhosts). After writing to httpd.conf, depending on the DB creation option it creates DB with provided hostname, db user and db password.

Example Usage:

```
my $vhosts = $apache->get_vhosts();
$apache->create_vhost($newopts, $vhosts);
```

delete_vhost(\$)

This function used to delete a virtual host entry from httpd.conf. Virtual hostname is passed as argument. Besides deleting virtual host from httpd.conf, it deletes whole directory tree related to that virtual host, it also drops related database, delete users, database permissions, disconnect to MySQL.

Example Usage:

```
$apache->delete_vhost($vhost);
```

update_vhost(\$)

This function used to update a virtual host entry in httpd.conf. It receives hash reference containing values of different virtual host parameters. It returns a string containing hostname-port format.

Example Usage:

```
$opt = { 'port' => 8000 };
my $newvhostkey = $apache->update_vhost($opt);
```

create_db(\$\$\$)

This function is used to create a MySQL Database with certain DB username and password. This function takes DB name, DB Username, DB password as arguments. It creates database, insert username and password in that as given with setting proper permissions.

Example Usage:

```
$apache->create_db($servername, $db_uname, $db_pass);
```

passchange_db(\$\$\$)

This function is used to change password of given user for a given database. It receives DB name, username and new password as arguments.

Example Usage:

```
$apache->passchange_db($servername, $db_uname, $db_pass);
```

set_docopts(\$)

This function receives document options as hash reference, depending on the options (e.g. includes, incnoexec etc) it generates a string and after checking some operations it writes them to httpd.conf.

Example Usage:

```
$opts = {
    'includes' => 'yes',
    'incnoexec' => 'no',
    'indexes' => 'no'
}
$apache->set_docopts($opts);
```

remove_alias(\$)

This function is used to remove an alias from httpd.conf. Hash reference containing alias to be deleted is passed as argument.

Example Usage:

```
$opts = {
    'vhost' => $vhost,
    'removealias' => $alias,
}
$apache->remove_alias($opts);
```

create_alias(\$)

This function is used to add an alias to httpd.conf. hash reference containing values of alias_to and alias_from is passed as arguments.

Example Usage:

```
$opts = {
    'vhost' => $vhost,
    'alias_from' => '/alias/from',
    'alias_to' => '/alias'
}
$apache->create_alias($opts);
```

remove_redirect(\$)

This function is used to remove a redirect from httpd.conf. Hash reference containing value of redirect(to be removed) is passed as argument.

Example Usage:

```
$opts = {  
    'vhost'          => $vhost,  
    'removedirect' => $redirect,  
}  
$apache->remove_redirect($opts);
```

create_redirect(\$)

This function is used to add a redirect to httpd.conf. Hash reference containing value of redirect_to and redirect_from is passed as arguments.

Example Usage:

```
$opts = {  
    'vhost'          => $vhost,  
    'redirect_from' => '/redirect/from',  
    'redirect_to'   => '/redirect/to'  
}  
$apache->create_redirect($opts);
```

save_errordocs(\$)

This function is used to save changed error documents for a given host into the httpd.conf. It receives hash reference containing various options containing virtual hostname and errordocuments with their values.

Example Usage:

```
$opts = {  
    'vhost' => $vhost,  
    'errordoc301' => /doc/301,  
    'errordoc302' => /doc/302,  
    .....  
}  
$apache->save_errordocs($opts);
```

find_vhost_db(\$)

This function is used to get DB username and DB name as hash reference for a virtual host. Virtual host name(server name) is passed as argument to the function.

Example Usage:

```
$dbref = $apache->find_vhost_db($servername);  
$DBuser = $dbref->{dbuser};  
$DBname = $dbref->{dbname};
```

read_cert(\$)

This function is used to read a certificate from a given host. Hash reference containing detail of a given virtual host is passed as an argument and hash reference containing values of all certificate parameters(e.g. fingerprint, startdate, enddate, country, state, city etc.) is return as result.

Example Usage:

```
my $vhref      = $apache->get_vhosts();
my $vhost      = $vhref->{$in{vhost}};
my $cert       = $apache->read_cert($vhost);
```

create_cert(\$\$)

This function basically creates a certificate for a given virtual host. It receives two arguments, hash reference containing virtual host details and a hash reference containing certificate parameter values(name, country, state, city, organization, department,email). Using these information it creates certificate and also modifies httpd.conf.

Example Usage:

```
my $vhref      = $apache->get_vhosts();
my $vhost      = $vhref->{$in{vhost}};
$apache->create_cert($vhost, $opts);
```

install_cert(\$\$)

This function is used to install certificate for a given virtual host. It receives two arguments, hash reference containing virtual host details and a hash reference containing values like certificate,key. Certificate is created using certificate file and key file. SSL option is taken into consideration.

Example Usage:

```
my $vhref      = $apache->get_vhosts();
my $vhost      = $vhref->{$in{vhost}};
$apache->install_cert($vhost, $opt);
```

generate_csr(\$\$)

This function is used to generate Certificate Signing Request. It receives two arguments, hash reference containing virtual host details and a hash reference containing certificate parameter values(name, country, state, city, organization, department,email).

Example Usage:

```
my $vhref      = $apache->get_vhosts();
my $vhost      = $vhref->{$in{vhost}};
my $tempcsr    = $apache->generate_csr($vhost, $opt);
```

validate_conf()

This function is used to validate/verify httpd.conf using /usr/sbin/httpd. It returns appropriate message on execution(Failed/Pass).

Example Usage:

```
my $validate = $apache->validate_conf();
```

restart()

This function is used to restart web server(httpd).

Example Usage:

```
$apache->restart();
```

3.1.5. AUTHORS

Pax Dickinson <pax@guardiandigital.com>

3.1.6. COPYRIGHT AND LICENSE

Copyright Guardian Digital, Inc., All Rights Reserved

3.2. datetime

3.2.1. NAME

datetime - Guardian Digital WebTool module interface to the system date and time.

3.2.2. SYNOPSIS

```
use datetime;  
my $time = new datetime;
```

3.2.3. DESCRIPTION

This Guardian Digital WebTool module provides an interface to manipulating the system date and time. This module allows you to manage the system NTP servers, set the system locale/timezone, and set the system date and time.

3.2.4. METHODS

new()

This constructor instantiates a new `datetime` object, which contains the NTP configuration and the zone configuration.

Example Usage:

```
my $time      = new datetime;

my $ntp1      = $time->{'ntp_config'}->{'server'}->[0];
my $ntp2      = $time->{'ntp_config'}->{'server'}->[1];
my $ntp3      = $time->{'ntp_config'}->{'server'}->[2];
my $region    = $time->{'zone_config'}->{'region'};
my $area      = $time->{'zone_config'}->{'area'}
```

restart_ntpd()

This function is used to restart NTP daemon if and only if its running.

Example Usage:

```
$time->restart_ntpd();
```

enumerate_ntp_servers()

This function can be used to get all listed ntp servers. It returns array of servers.

Example Usage:

```
@servers = $time->enumerate_ntp_servers();
```

enumerate_zoneinfo()

This function serves purpose of getting all zone and area information by reading directory tree `/usr/share/zoneinfo`. It makes a hash containing zones as keys and areas as their values and returns a reference.

Example Usage:

```
$zones = $time->enumerate_zoneinfo();
```

set_ntp_servers(\$\$\$)

This function is used to set all three ntp servers in configuration file. It receives three server names as argument, then it updates `/etc/ntp/ntp.conf`. It also writes all server names in `/etc/ntp/step-tickers` too.

Example Usage:

```
$time->set_ntp_servers($server1, $server2, $server3);
```

set_system_time(\$)

This function sets the system time according to given argument. Hash reference containing values of month,day,year,hour,minute,second is received as argument.

Example Usage:

```
$time->set_system_time({
    'month'      => $month,
    'day'        => $day,
    'year'       => $year,
    'hour'       => $hour,
    'minute'    => $minute,
    'second'    => $second,
});
```

set_zoneinfo(\$\$)

This function is used to set zone information. Region and area are passed as argument, so old symbolic link to /etc/localtime is deleted and new link is created according to /usr/share/zoneinfo/region/area.

Example Usage:

```
$time->set_zoneinfo($zone, $area);
```

3.2.5. AUTHORS

Ryan W. Maple <ryan@guardiandigital.com>

3.2.6. COPYRIGHT AND LICENSE

Copyright Guardian Digital, Inc., All Rights Reserved

3.3. firewall

3.3.1. NAME

firewall - Guardian Digital WebTool module interface to Shorewall firewall.

3.3.2. SYNOPSIS

```
use firewall;
my $firewall = new firewall;
```

3.3.3. DESCRIPTION

This Guardian Digital WebTool module provides an interface to manipulating the system firewall. This module allows you manage the general firewall configuration, parameters (host/network aliases), the blacklist, port forwarding, network interfaces and logical zones, policies, and, of course, firewall rules.

3.3.4. METHODS

is_installed()

This function is used to check shorewall is installed or not.

Example Usage:

```
if (!$firewall->is_installed()){
.....
}
```

blacklist_create(\$)

This function is used to create an entry in firewall blacklist (/etc/shorewall/blacklist). Hash reference with information containing address, protocol and ports is passed as argument.

Example Usage:

```
my $b = {
    'address'      => $address,
    'protocol'     => $protocol,
    'ports'        => $ports
};
$firewall->blacklist_create($b);
```

blacklist_delete(\$)

This function is used to delete a specified entry from firewall blacklist (/etc/shorewall/blacklist). Key is passed as an argument, which can be obtained by calling list_blacklist subroutine.

Example Usage:

```
my $blacklist = $firewall->list_blacklist();
foreach my $k (sort keys %{$blacklist}) {
    my $b      = $blacklist->{$k};
    $key       = $b->{'key'};
    $firewall->blacklist_delete($key);
}
```

get_blacklist_entry(\$)

This function is used to get a blacklist entry detail(address,ports,ports_type,protocol, protocol,protocol_type,key) by passing a key as argument, which can be obtained by calling list_blacklist subroutine. Hash reference is received in which above parameters are as keys and their respective values as hash values.

Example Usage:

```
my $blacklist = $firewall->list_blacklist();
foreach my $k (sort keys %{$blacklist}) {
    my $b = $blacklist->{$k};
    $key = $b->{'key'};
    $blacklist_entry = $firewall->get_blacklist_entry($key);
}
```

param_create(\$)

This function is used to create an entry in param configuration file /etc/shorewall/param (Hosts and Networks). It receives a hash reference as an argument containing values of name,zone and addresses.

Example Usage:

```
my $p = {
    'name'      => $name,
    'zone'      => $zone,
    'addresses' => \@addresses,
};
$firewall->param_create($p);
```

param_delete(\$)

This function is used to delete a specified entry from /etc/shorewall/param. Key is passed as an argument, which can be obtained by calling list_params subroutine.

Example Usage:

```
my $param = $firewall->list_params();
foreach my $k (sort keys %{$param}) {
    my $b = $param->{$k};
    $key = $b->{'key'};
    $firewall->param_delete($key);
}
```

get_param_entry(\$)

This function is used to get a param entry detail(name,zone,key) as hash reference by passing a key as an argument, which can be obtained by calling list_params subroutine.

Example Usage:

```
my $param = $firewall->list_params();
foreach my $k (sort keys %{$param}) {
```

```

my $b    = $param->{$k};
$key     = $b->{'key'};
$param_entry = $firewall->get_param_entry($key);
}

```

portfwd_create(\$)

This function is used to create a port forwarding rule(making entry in /etc/shorewall/ rules). It receives a hash reference as argument(see example).

Example Usage:

```

my $r = {
    'action'      => $in{'action'},
    'src'         => $in{'src'},
    'src_zone'    => $in{'src_zone'},
    'src_port'    => $in{'src_port'},
    'dst'         => $in{'dst_address'},
    'dst_zone'    => $in{'dst_zone'},
    'dst_port'    => $in{'dst_port'},
    'protocol'    => $in{'protocol'},
};
$firewall->portfwd_create($r);

```

rule_create(\$)

This function is used to create a rule(ACCEPT/REJECT - making entry in /etc/shorewall/ rules). It receives a hash reference as argument(see example).

Example Usage:

```

my $r = {
    'action'      => $in{'action'},
    'src'         => $in{'src'},
    'dst'         => $in{'dst'},
    'protocol'    => $in{'protocol'},
    'dst_port'    => \@dst_port,
    'src_port'    => \@src_port,
};
$firewall->rule_create($r);

```

rule_delete(\$)

This function is used to delete a rule from /etc/shorewall/rules. It receives a key as an argument to delete a specified rule, which can be obtained by calling list_rules subroutine.

Example Usage:

```

my $rules = $firewall->list_rules();
foreach my $k (sort keys %{$rules}) {
    my $b    = $rules->{$k};
    $key     = $b->{'key'};
}

```

```
$firewall->rule_delete($key);  
}
```

get_rule_entry(\$)

This function is used to get a rule entry detail(action,source zone,protocol) by passing a key as an argument, which can be obtained by calling list_rules subroutine.

Example Usage:

```
my $rules = $firewall->list_rules();  
foreach my $k (sort keys %{$rules}) {  
    my $b = $rules->{$k};  
    $key = $b->{'key'};  
    $rule_entry = $firewall->get_rule_entry($key);  
}
```

list_blacklist()

This function reads through the file /etc/shorewall/blacklist and makes a hash reference of hash, so that each address as key will contain address, protocol, ports, key, protocol type, ports type as its value. It returns hash reference as a result.

Example Usage:

```
my $blacklist = $self->list_blacklist();  
foreach my $b (sort keys %{$blacklist}) {  
    my $address = $blacklist->{$b}->{'address'};  
    my $protocol = uc($blacklist->{$b}->{'protocol'});  
    my $ports = $blacklist->{$b}->{'ports'};  
    my $key = $blacklist->{$b}->{'key'};  
    my $ports_type = $blacklist->{$b}->{'ports_type'};  
    my $protocol_type = $blacklist->{$b}->{'protocol_type'};  
}
```

list_interfaces()

This function basically reads through the shorewall interface configuration file (/etc/shorewall/interfaces) and returns hash reference as a result(see example for hash reference detail). Each interface as key will contain interface, broadcast, lineno, options as its value.

Example Usage:

```
my $interfaces = $self->list_interfaces();  
foreach my $i (sort keys %{$interfaces}) {  
    my $zone = $interfaces->{$i}->{'zone'};  
    my $interface = uc($interfaces->{$i}->{'interface'});  
    my $broadcast = $interfaces->{$i}->{'broadcast'};  
    my $lineno = $interfaces->{$i}->{'lineno'};  
    my $options1 = $interfaces->{$i}->{'options'}->{'options1'};  
    my $options2 = $interfaces->{$i}->{'options'}->{'options2'};  
    .....  
}
```

```
}
```

list_masq(\$)

This function basically reads through the shorewall masquerading configuration file (/etc/shorewall/masq) and makes a hash containing values like source interface, destination interface, enable/disable. It returns hash reference as a result(see example).

Example Usage:

```
my $masq = $self->list_masq();
$enable_status = $masq->{$dst_interface}->{$src_interface};
```

list_options(\$)

This function used to returns a hash reference of translations currently defined in WebTool. currently those are dhcp, norfc1918, blacklist, tcpflags, routeback.

Example Usage:

```
$options = $firewall->list_options($page);
$dhcp_translated = $options->{'dhcp'};
$blacklist_translated = $options->{'blacklist'};
```

list_params()

This function basically reads through the shorewall param configuration file (/etc/shorewall/param) and it returns hash reference as a result. Each param as key will contain name, zone, addresses, key as its values. see example for detail.

Example Usage:

```
my $params = $self->list_params();
foreach my $b (sort keys %{$params}) {
    my $name = $params->{$b}->{'name'};
    my $zone = $params->{$b}->{'zone'};
    my $addresses = $page->str_truncate(join(',', @{$params->{$b}->{'addresses'}}), 30);
    my $key = $params->{$b}->{'key'};
}
```

list_policy()

This function reads through the shorewall policy configuration file (/etc/shorewall/policy) and makes a hash containing values like source,destination, action. It returns hash reference as a result.

Example Usage:

```
my $policy = $firewall->list_policy();
$action = $policy->{$dst}->{$src};
```

list_ports()

This function can be used to get a list of all ports in hash reference. In hash, Keys will have port names and value will be their description.

Example Usage:

```
my $ports = $firewall->list_ports();
```

list_rules()

This function reads through the shorewall rule configuration file (/etc/shorewall/rules) and makes a hash containing values like action, source, destination, protocol, port etc. It returns hash reference as a result.

Example Usage:

```
my $rules = $self->list_rules();
foreach my $r (@{$rules}) {
    my $action      = $r->{'action'};
    my $src         = $r->{'src'};
    my $src_zone    = $r->{'src_zone'};
    my $src_ports   = join(',', @{$r->{'src_ports'}});
    my $dst         = $r->{'dst'};
    my $dst_zone    = $r->{'dst_zone'};
    my $dst_ports   = join(',', @{$r->{'dst_ports'}});
    my $protocol    = uc($r->{'protocol'});
    my $key         = $r->{'key'};
    my $orig_dest   = $r->{'orig_dest'};
    my $sort        = $r->{'sort'};
    my $raw         = $r->{'raw'};
}
```

list_zones()

This function reads through the shorewall zones configuration file (/etc/shorewall/zones) and returns hash reference as a result. Each zone as key will contain zone, display, comment as its values. see example for detail.

Example Usage:

```
my $zones = $self->list_zones();
foreach my $b (sort keys %{$zones}) {
    my $name        = $zones->{$b}->{'zone'};
    my $zone        = $zones->{$b}->{'display'};
    my $comment     = $zones->{$b}->{'comment'};
}
```

write_interfaces(\$)

This function receives a array reference of hash containing interface, zone and options. For every interface it makes comma separated list for options. Then it writes out all information in interfaces configuration file (/etc/shorewall/interfaces).

Example Usage:

```
push @interfaces, {
  'interface' => $interface,
  'zone'      => $zone,
  'options'   => $options->{$interface},
}
$firewall->write_interfaces(\@interfaces);
```

write_masq(\$)

This function receives a array reference of hash containing masq information like destination interface and source interface. Then it writes out all information in masq configuration file(/etc/shorewall/masq).

Example Usage:

```
push @masq, {
  'src_interface' => $src_interface,
  'dst_interface' => $dst_interface,
}
$firewall->write_masq(\@masq);
```

write_policy(\$)

This function receives a array reference of hash containing source, destination and action. Depends on the action(ACCEPT/REJECT) in policy it writes out all information in policy configuration file (/etc/shorewall/policy).

Example Usage:

```
push @policy, {
  'src'      => $src,
  'dst'      => $dst,
  'action'   => $action
};
$firewall->write_policy(\@policy);
```

enable_shorewall()

This function can be used to enable shorewall firewall at start up. It edits shorewall configuration file(/etc/shorewall/shorewall.conf).

Example Usage:

```
$firewall->enable_shorewall();
```

3.3.5. AUTHORS

Ryan W. Maple <ryan@guardiandigital.com>

3.3.6. COPYRIGHT AND LICENSE

Copyright Guardian Digital, Inc., All Rights Reserved

3.4. ftp

3.4.1. NAME

ftp - Guardian Digital WebTool module interface to the vsftpd FTP server.

3.4.2. SYNOPSIS

```
use ftp;  
my $ftp = new ftp($page);
```

3.4.3. DESCRIPTION

This Guardian Digital WebTool module provides an interface to manipulating the system FTP server. This module allows you to manage the vsftpd configuration file, chroot list, and blacklist.

3.4.4. METHODS

read_vsftpd_xinetd_conf()

This function reads `/etc/xinetd.d/ftp` configuration file line by line and returns interface it should listen to, in form of hash reference.

Example Usage:

```
my $xinetd = $ftp->read_vsftpd_xinetd_conf();  
my $interface = $xinetd->{'iface'};
```

write_vsftpd_xinetd_conf(\$)

This function basically writes interface that it should listen to, into `/etc/xinetd.d/ftp` configuration file. It receives interface value as argument(hash reference), if interface is not defined than it sets 127.0.0.1(localhost) as default interface.

Example Usage:

```
$conf = { 'iface' => $interface };  
$ftp->write_vsftpd_xinetd_conf($conf);
```

restart()

This function restarts the xinetd service.

Example Usage:

```
$ftp->restart();
```

read_vsftpd_conf()

This function basically reads /etc/vsftpd.conf file line by line and makes hash of key-value pairs. It also reads contents of ftp's blacklist and chroot from their respective configuration files and updates the hash. Ultimately it returns hash reference containing all values as a result.

Example Usage:

```
my $conf = $ftp->read_vsftpd_conf();
```

write_vsftpd_conf(\$)

This function basically writes to /etc/vsftpd.conf file. It receives hash reference containing key-value pair(parameters as keys and their respective values) for vsftpd.conf file. It also writes interface value in ftp configuration file under xinetd.d directory.

Example Usage:

```
$ftp->write_vsftpd_conf($conf);
```

write_vsftpd_black_list(\$)

This function writes out users provided as argument(array reference), to /etc/ftpusers file(blacklist).

Example Usage:

```
@black_list = ('mysql','nobody');
$ftp->write_vsftpd_black_list(\@black_list);
```

write_vsftpd_chroot_list(\$)

This function writes out users provided as argument(array reference), to /etc/vsftpd.chroot_list file(chroot).

Example Usage:

```
@chroot_list = ('mysql','nobody');
$ftp->write_vsftpd_chroot_list(\@chroot_list);
```

3.4.5. AUTHORS

Ryan W. Maple <ryan@guardiandigital.com>

3.4.6. COPYRIGHT AND LICENSE

Copyright Guardian Digital, Inc., All Rights Reserved

3.5. libwrap

3.5.1. NAME

libwrap - Guardian Digital WebTool module interface to TCP Wrappers.

3.5.2. SYNOPSIS

```
use libwrap;  
my $libwrap = new libwrap;
```

3.5.3. DESCRIPTION

This Guardian Digital WebTool module provides an interface to manipulating TCP wrappers (/etc/hosts.allow and /etc/hosts.deny). This module allows you to add, delete, get, and list entries in the hosts.allow file.

3.5.4. METHODS

exists(\$\$\$)

This function is basically used to check out that particular entry(host) is exists in configuration file(hosts.allow/hosts.deny) or not? It returns true or false respectively. It takes service, host and file as arguments.

Example Usage:

```
$libwrap->exists($service, $host, $file);
```

resolve_service(\$)

This function is used to get service name based on the language selection, usually for display purpose on page. It receives two arguments one is WebTool::UI object and another is service name. It returns translated service name.

Example Usage:

```
$libwrap->resolve_service($page, 'pptpd')
```

allow_add(\$)

This function is used to add host to hosts.allow file. It receives service name and host(ip address or ALL) as arguments and update hosts.allow file by adding entry.

Example Usage:

```
$libwrap->allow_add($service, $host);
```

allow_delete(\$)

This function is used to delete host from hosts.allow file. It receives service name and host(ip address or ALL) as arguments and update hosts.allow file by deleting entry.

Example Usage:

```
$libwrap->allow_delete($service, $host);
```

allow_get()

This function is used to get information from configuration file(usually /etc/hosts.allow). It returns hash reference containing services as keys and allowed hosts as respective values (in terms of array reference).

Example Usage:

```
$allow = libwrap::allow_get();
foreach my $service (sort keys %{$allow}) {
    my @hosts = @{$allow->{$service}};
    ...
}
```

3.5.5. AUTHORS

Ryan W. Maple <ryan@guardiandigital.com>

3.5.6. COPYRIGHT AND LICENSE

Copyright Guardian Digital, Inc., All Rights Reserved

3.6. logging**3.6.1. NAME**

logging - Guardian Digital WebTool module interface to system and application logs.

3.6.2. SYNOPSIS

```
use logging;  
my $logging = new logging($page);
```

3.6.3. DESCRIPTION

This Guardian Digital WebTool module provides an interface to listing, retrieving, and searching system and application logs.

3.6.4. METHODS

get_system_logs()

This function returns hash reference(name,path) of system log(11), kernel messages(12), webtool audit(13) and user audit logs(14). See example for understanding number notations.

Example Usage:

```
my $logref = $logging->get_system_logs();  
my $syslog = $logref->{11}->{'name'};  
my $syspath = $logref->{11}->{'path'};
```

get_app_logs()

This function returns hash reference(name,path) of application log files like Mail server log(11), PHP error log(12), MySQL error log(13). see example for number notations.

Example Usage:

```
my $logref = $logging->get_app_logs();  
my $maillog = $logref->{11}->{'name'};  
my $mailpath = $logref->{11}->{'path'};
```

get_web_logs()

This returns hash reference(name,path) of all website log files. It basically reads through all virtual hosts hosted in web server file and depends on the port it selects which error log file it should choose(e.g port 443 :SSL port).

Example Usage:

```
my $logref = $logging->get_web_logs();  
my $weblog = $logref->{11}->{'name'};  
my $webpath = $logref->{11}->{'path'};
```

read_log(\$)

This function is used to read last 50 lines of the logfile. Logfile name is provided in as an argument.

Example Usage:

```
@logdata = $logging->read_log($file);
```

search_log(\$\$)

This function is used to search a specific pattern in last 50 lines of the logfile. Logfile name and search pattern is provided as arguments.

Example Usage:

```
@logdata = $logging->search_log($file, $search);
```

3.6.5. AUTHORS

Ryan W. Maple <ryan@guardiandigital.com>

3.6.6. COPYRIGHT AND LICENSE

Copyright Guardian Digital, Inc., All Rights Reserved

3.7. named**3.7.1. NAME**

named - Guardian Digital WebTool module interface to the BIND DNS server.

3.7.2. SYNOPSIS

```
use named;
my $named = new named($page);
```

3.7.3. DESCRIPTION

This Guardian Digital WebTool module provides an interface to manipulating the BIND DNS server. This module allows you to manage forward and reverse master and slave zones and zone records such as A (address), CNAME (name alias), MX (mail exchanger), NS (nameserver), and PTR (reverse addresses).

3.7.4. METHODS

is_installed()

This function is used to check DNS is installed or not.

Example Usage:

```
if (!$named->is_installed()) ...
```

inaddr2foo(\$)

This subroutine takes an in-addr.arpa address as an argument and turns it into the IP address.

Example Usage:

```
$named->inaddr2foo($zone);
```

reload()

This function checks to see if named service is running, if its running than it reloads else it returns zero.

Example Usage:

```
$named->reload();
```

restart()

This function checks to see if named service is running, if its running than it restarts else it returns zero.

Example Usage:

```
$named->restart();
```

read_named_conf()

This function processes and returns the named.conf file in a complex hash reference. For details on the return value you should look at the file.

Example Usage:

```
my $named_conf = $named->read_named_conf();
```


write_named_conf(\$)

Basic function is to write(edit) a named.conf file. It receives a hash reference of array reference of configuration parameters(allow-query,allow-transfer,listen-on etc). It reads through the named.conf file and compare parameters and values with provided parameters and ultimately updates/writes named.conf file.

Example Usage:

```
my $cfg = {
    'allow-query'           => \@allow_query,
    'allow-transfer'       => \@allow_transfer,
    'forwarders'           => \@forwarders,
    'listen-on'            => \@listen_on,
    'forward-only'         => ($forwarders[0] ? 1 : 0,
};
$named->write_named_conf($cfg);
```

zone_generate_email(\$)

This function receives an email as argument, it substitute '@' with '.', and '.' is attached at the end of string if it doesnt have it.

Example Usage:

```
my $email = $named->zone_generate_email($email);
```

zone_generate_filename(\$\$)

This function receives zone type and origin as arguments and it returns string like this master/db.' . \$origin(generating file under master directory).

Example Usage:

```
my $filename = $named->zone_generate_filename($type, $origin);
```

zone_generate_origin(\$\$)

This function receives zone type and name as arguments. If type is 'forward' and name doesnt end with '.' then name is appended with '.'. If type is 'reverse' then name is split with '.' and splitted string is concated with '.', and whole string now is concated with 'in-addr.arpa.' and returned.

Example Usage:

```
my $origin = $named->zone_generate_origin($type, $name);
```

zone_exists(\$\$)

This function checks existance of zone in named.conf. It receives zone type and name as arguments. It looks for given type and name in named.conf, if it find this pair then it returns 1 else it returns 0.

Example Usage:

```
$exists = $named->zone_exists('master', $origin);
```

zone_create(\$)

This function basically creates a master zone in named.conf. It receives hash reference containing zone type, name, email as argument. Using these information it creates origing, email and filename values.Finally it writes out these information in named.conf.

Example Usage:

```
my $zone = {
    'TTL'                => $in{'ttl_a'} . $in{'ttl_b'},
    'allow-query'        => \@allow_query,
    'allow-transfer'     => \@allow_transfer,
    'type'               => $in{'type'},
    'master'             => $in{'master'},
    'email'              => $in{'email'},
    'name'               => $in{'zone_name'},
};
$create = $named->zone_create($zone);
```

zone_update(\$)

This function basically updates given zone. It receives hash reference containing zone type, name, email as argument. Using zone type it generates email specific to that zone. Finally it updates named.conf and zone file too.

Example Usage:

```
my $zone = {
    'TTL'                => $in{'ttl_a'} . $in{'ttl_b'},
    'allow-query'        => \@allow_query,
    'allow-transfer'     => \@allow_transfer,
    'type'               => $in{'type'},
    'master'             => $in{'master'},
    'email'              => $in{'email'},
    'name'               => $in{'zone_name'},
};
$name->zone_update($zone);
```

zone_delete(\$)

This function is used to delete a specified zone. It receives hash reference containing zone name as argument. It reads through named.conf file and deletes that zone information. Zone file is also deleted from respective directory(master/slave).

Example Usage:

```
$named->zone_delete($zone);
```

zone_get(\$)

This function retrieves the specified zone, including all data from the named.conf and the records/SOA from the zone file. It receives a hash reference of zone information as argument.

Example Usage:

```
my $zone_data = $named->zone_get($zone);
```

zone_create_slave(\$)

This function is used to create a slave zone. It receives hash reference containing zone information as argument. It writes out these information in named.conf. It also creates zone file at appropriate location(slave/).

Example Usage:

```
my $zone = {
    'allow-query'           => \@allow_query,
    'allow-transfer'       => \@allow_transfer,
    'masters'              => \@masters,
    'type'                  => $in{'type'},
    'name'                  => $in{'zone_name'},
};
$name->zone_create_slave($zone);
```

zone_update_slave(\$)

This function is used to update slave zone. It receives hash reference containing zone information as argument. It updates these information in named.conf.

Example Usage:

```
my $zone = {
    'allow-query'           => \@allow_query,
    'allow-transfer'       => \@allow_transfer,
    'masters'              => \@masters,
    'name'                  => $in{'zone_name'},
};
$name->zone_update_slave($zone);
```

bump_serial(\$)

This function is used to change provided(argument) zonefile. It reads through zone file, and it updates a line with timestamp, it just increases the serial number after the time stamp by 1.

Example Usage:

```
$named->bump_serial($zone);
```

record_clean(\$\$)

This function receives two arguments. First argument is type(rval or lval). If type will be rval and second argument will not end with ':' then it will just append ':' at the end in second argument and return it. If type will be lval and second argument ends up with ':' then it splits second argument with ':' and returns first element of second argument.

Example Usage:

```
my $master = $named->record_clean('rval', $zone->{'master'});
```

record_create(\$@)

This function basically creates a record in zone file. It receives two arguments. First argument is a hash reference containing record data and the second (optional) argument (yes or no, default is yes) dictates whether or not the zones serial number is updated.

Depending on the record type (A,CNAME,MX,NS etc) it writes out record details in appropriate zonefile.

Example Usage:

```
my $r = {
    'zone'      => $in{'zone'},
    'type'      => $in{'type'},
    'left'      => $in{'field1'},
    'right'     => $in{'field2'},
    'weight'    => $in{'field2'},
};
$named->record_create($r);
```

record_delete(\$@)

This function basically deletes a record from a zone file. It receives two arguments. First argument is a hash reference containing record data and the second (optional) argument (yes or no, default is yes) dictates whether or not the zones serial number is updated.

Depending on the record type (A,CNAME,MX,NS etc) it deletes record details from zone file.

Example Usage:

```
my $r = {
    'zone'      => $in{'zone'},
    'type'      => $in{'type'},
    'left'      => $in{'field1'},
    'right'     => $in{'field2'},
    'weight'    => $in{'field2'},
};
$named->record_delete($r);
```

3.7.5. AUTHORS

Ryan W. Maple <ryan@guardiandigital.com>

3.7.6. COPYRIGHT AND LICENSE

Copyright Guardian Digital, Inc., All Rights Reserved

3.8. network

3.8.1. NAME

network - Guardian Digital WebTool module interface to networking.

3.8.2. SYNOPSIS

```
use network;
my $network = new network();
```

3.8.3. DESCRIPTION

This Guardian Digital WebTool module provides an interface to manipulating networking. This module allows you to manage physical interfaces, virtual interfaces, /etc/host entries, the system hostname, nameservers in /etc/resolv.conf, and static/default routes.

3.8.4. METHODS

new()

This constructor sets initial values for parameters. It includes getting host name from system, listing hosts which contains IP addresses as keys and hostname as their respective values and from resolv.conf hash containing key value pair, where value will be array reference of all values corresponding to particular key. First key-value pair will be nameserver and ip address pair.

Example Usage:

```
my $network = new network();
$hostname   = $network->{'conf_hostname'};
$host1      = $network->{'conf_hosts'}->{$ip_address1};
$host2      = $network->{'conf_hosts'}->{$ip_address2};
$ipaddr     = $network->{'conf_resolv'}->{$nameserver};
```

host_add(\$)

This function is used to add an entry(host) to /etc/hosts. It receives a hash reference containing IP address and domain name of host.

Example Usage:

```
$host = {  
    'address' => $address,  
    'value'   => $value  
}  
$network->host_add($host);
```

host_delete(\$)

This function is used to delete an entry(host) from /etc/hosts. It receives IP address of host to be deleted as argument.

Example Usage:

```
$network->host_delete($hostip);
```

host_get(\$)

This function is used to get host information as key-value pairs(address and hostname) from /etc/hosts. It receives IP address of host as argument.

Example Usage:

```
$network->host_get($host);  
$address    = $host->{'address'};  
$hostname   = $host->{'value'};
```

interface_create(\$)

Basically this function creates an entry in /etc/network/interfaces. It receives interface hash reference which contains information like type, address and netmask. Depending on the type(static,dynamic,virtual) network and broadcast address is generated and all these information is written into configuration file. And after that interface is brought up.

Example Usage:

```
if ($type eq 'static') {  
    $i->{'address'} = $in{'static_address'};  
    $i->{'netmask'} = $in{'static_netmask'};  
}  
elsif ($type eq 'virtual') {  
    $i->{'address'} = $in{'virtual_address'};  
    $i->{'netmask'} = $in{'virtual_netmask'};  
    $i->{'attach'} = $in{'input_virtual_attach'};  
}  
my $iface = $network->interface_create($i);
```

interface_delete(\$)

Basically this function is used to delete an interface entry from /etc/network/interfaces file. It receives interface name as argument.

Example Usage:

```
$network->interface_delete($interface);
```

interface_get(\$)

This function is used to get an detail information about some interface. Interface name is passed as argument to this function.

Example Usage:

```
my $interface = $self->interface_get($interface);
```

interfaces_list()

This function reads through the /etc/network/interfaces file, and returns a hash reference of all interface details depending on their properties.

Example Usage:

```
my $interfaces = $network->interfaces_list();
foreach my $if (sort keys %{$interfaces}) {
    my $iface    = $interfaces->{$if}->{'iface'};
    my $address  = $interfaces->{$if}->{'address'};
    my $netmask  = $interfaces->{$if}->{'netmask'};
    my $type     = $interfaces->{$if}->{'type'};
}
```

default_route_create(\$)

This function is used to create a default route, add an entry in /etc/network/interfaces file, it receives route as hash reference containing information like device and default gateway. see example.

Example Usage:

```
$network->default_route_create({
    'device'      => $in{'default_device'},
    'gateway'     => $in{'default_gateway'},
});
```

default_route_delete(\$)

This function is used to delete default route entry from /etc/network/interfaces file, it receives device(interface) name as argument.

Example Usage:

```
$network->default_route_delete($default_device);
```

route_create(\$)

This function is used to add static route to /etc/network/interfaces file, it receives hash reference containing device and gateway values as argument.

Example Usage:

```
my $i = {  
    'device'      => $in{'device'},  
    'network'    => $in{'network'},  
    'netmask'    => $in{'netmask'},  
    'gateway'    => ($in{'gateway'}) ? $in{'gateway'} : undef,  
};  
$network->route_create($i);
```

route_delete(\$\$\$)

This function is used to delete a static route from /etc/network/interfaces file, it receives three arguments- device, network and netmask.

Example Usage:

```
$network->route_delete($device, $network, $netmask);
```

route_get(\$\$\$)

This function is used to get a static route information from /etc/network/interfaces file, it receives three arguments- device, network and netmask.

Example Usage:

```
$network->route_get($device, $network, $netmask);
```

ifdown(\$)

This function is used to bring down the provided interface.

Example Usage:

```
$postfix->ifdown($iface);
```


ifup(\$)

This function is used to bring up the provided interface.

Example Usage:

```
$postfix->ifup($iface);
```

restart_networking()

This function is used restart the network service.

Example Usage:

```
$network->restart_networking();
```

make_broadcast(\$)

This function is used to get broadcast address. It requires ip address and netmask to be passed as arguments.

Example Usage:

```
my $broadcast = make_broadcast($address, $netmask);
```

make_network(\$)

This function is used to get network address. It requires ip address and netmask to be passed as arguments.

Example Usage:

```
my $network = make_network($address, $netmask);
```

write_pump_conf(\$)

This function writes /etc/pump.conf file. If argument received is not null then it writes 'nodns' in file.

Example Usage:

```
write_pump_conf($dynamic_dns);
```

set_hostname(\$)

This function sets the hostname of the system. It receives hostname as argument.

Example Usage:

```
$network->set_hostname($hostname);
```

set_resolv(\$)

This function receives hash reference containing values of search and servers. Ultimately it writes out those values in /etc/resolv.conf. See example.

Example Usage:

```
$network->set_resolv({
    'search'      => \@dns_search,
    'servers'     => \@dns_servers,
});
```

mask2cidr(\$)

This function converts netmask to cidr. It receives netmask as argument.

Example Usage:

```
my $cidr = $network->mask2cidr($netmask);
```

cidr2mask(\$)

This function converts cidr to netmask. It receives cidr as argument.

Example Usage:

```
my $netmask = $network->cidr2mask($cidr);
```

3.8.5. AUTHORS

Ryan W. Maple <ryan@guardiandigital.com>

3.8.6. COPYRIGHT AND LICENSE

Copyright Guardian Digital, Inc., All Rights Reserved

3.9. postfix

3.9.1. NAME

postfix - Guardian Digital WebTool module interface to the Postfix SMTP server.

3.9.2. SYNOPSIS

```
use postfix;
my $postfix = new postfix;
```

3.9.3. DESCRIPTION

This Guardian Digital WebTool module provides an interface to manipulating the Postfix mail server. This module allows you to manage aliases, transports (mail routes), and virtual domains.

3.9.4. METHODS

is_installed()

This function is used to check postfix is installed or not.

Example Usage:

```
if (!$postfix->is_installed()){
    .....
}
```

reload()

This function is used to reload the postfix(/etc/init.d/postfix) service.

Example Usage:

```
$postfix->reload();
```

readval(\$)

This function is used to get a specified configuration parameter's(argument) value of postfix mail system.

Example Usage:

```
$mailboxsizelimit = $postfix->readval("mailbox_size_limit");
```

readsmtp()

This function is used to determine if smtpd is enabled. If its enabled then returns 'yes' else 'no'.

Example Usage:

```
$enable_status = $postfix->readsmtp();
```

updatesmtp(\$)

This function is used to activate or deactivate smtpd service. It receives one argument, if argument is 'yes' then it will activate smtpd, if argument is 'no' it will deactivate smtpd.

Example Usage:

```
updatesmtp('yes');
```

setval(\$\$)

This function is used to set specified option's value using postconf(/usr/sbin/postconf). It receives two arguments(option and value).

Example Usage:

```
setval(undef, 'message_size_limit', 3200);
```

smtpd_rr_util(\$\$)

This function inserts or removes a given entry from the Postfix smtpd_recipient_restrictions. The first argument is 'insert' or 'remove',and the second argument is the restriction you want to insert or remove.

Example Usage:

```
smtpd_rr_util(undef, 'insert', 'reject_unknown_sender_domain');
```

get_smtpd_rr_val()

This function reads value for restrictions of smtpd recipients, and depends on checking conditions it returns value 'strict', 'moderate' or 'off'.

Example Usage:

```
$restriction = $postfix->get_smtpd_rr_val();
```

config_write(\$)

This function writes a configuration depends on the user selected options. It receives hash reference containing all values(e.g. relayhost, enable-disable procmail, queue lifetime etc). Using setval function it sets all values according to options. additionally depends on client restriction it applies suitable operations (insert,remove)too.

Example Usage:

```
$postfix->config_write($newopts);
```

get_domains()

This function reads through the /etc/postfix/virtual file, and returns an array of virtual mail domains.

Example Usage:

```
my @virtual_domains = $postfix->get_domains();
```

add_domains(\$\$)

This function allows us to add an virtual mail domain by passing domain name and postmaster as argument.

Example Usage:

```
$postfix->add_domain($domain_name, $postmaster);
```

delete_domain(\$)

This function can be used to delete a virtual domain. Domain name is passed as an argument. It deletes corresponding relay domain too.

Example Usage:

```
$postfix->delete_domain($domain_name);
```

get_relaydomains()

This function can be used to get a array of relay domains. It filters out \$mydestination (list of domains that this machine considers itself the final destination for)and \$transport_maps.

Example Usage:

```
my @relays = get_relaydomains();
```

save_relaydomains(@)

This function receives an array of relay domains and save them(writes out in main.cf).

Example Usage:

```
save_relaydomains(@newrelays);
```

add_relaydomain(\$)

This function add a given relay domain to the pool of relay domains. It receives a domain name as an argument.

Example Usage:

```
add_relaydomain($domain_name);
```

delete_relaydomain(\$)

This function deletes a given relay domain. It receives a domain name as an argument.

Example Usage:

```
delete_relaydomain($domain_name);
```

get_mappings(\$)

This function can be used to get list of all addresses in a virtual domain(address, Recipient). It receives a domain name as an argument. It returns array of hash references. see example for detail.

Example Usage:

```
my @addresses = $postfix->get_mappings($domain);
$addresses[$j]->{index} = $index;
$addresses[$j]->{address} = $address;
$addresses[$j]->{recipient} = $recipient;
```

add_mapping(\$\$\$)

This function can be used to add an email(address,recipient) to a virtual domain. It receives a domain name, email address, and recipient as arguments.

Example Usage:

```
$postfix->add_mapping($domain_name, $email_address, $recipient);
```

delete_mapping(\$\$\$)

This function can be used to delete an email(address,recipient) from a virtual domain. It receives a domain name, address, and recipient as arguments.

Example Usage:

```
$postfix->delete_mapping($domain_name, $email_address, $recipient);
```

update_mapping(\$\$\$)

This function is used to update an email(address,recipient) in a virtual domain. It receives a domain name, address, and recipient as arguments.

Example Usage:

```
$postfix->update_mapping($domain_name, $email_address, $recipient);
```

rebuild_virtual_map()

This function can be used to rebuild virtual map in hash file using postmap utility.

Example Usage:

```
$postfix->rebuild_virtual_map();
```

parse_aliases(\$)

This function returns a hash reference containing all of the mail aliases that currently exist on the system. By default it will skip over "system" aliases that the end-user should never manipulate, but you can tell it to pass `_all_` aliases back up by passing a single argument of 1 (true).

Example Usage:

```
my $aliases = $postfix->parse_aliases(1);
```

add_alias(\$\$)

This function can be used to add an alias to aliases configuration file, alias and recipient values are passed as arguments.

Example Usage:

```
$postfix->add_alias($alias, $recipient);
```

update_alias(\$)

This function is used to update an alias entry in aliases configuration file, alias and recipient values are passed as arguments.

Example Usage:

```
$postfix->update_alias($alias, $recipient);
```

delete_alias(\$)

This function is used to delete an alias entry from aliases configuration file, alias value is passed as arguments.

Example Usage:

```
$postfix->delete_alias($alias);
```

get_alias_recipient(\$)

This function is used to get recipient for particular alias. Alias is passed as an argument. it returns corresponding recipient.

Example Usage:

```
my $recipient = $postfix->get_alias_recipient($alias);
```

resolve_alias(\$\$)

This function takes an alias and a method (files) as its arguments and returns what the alias resolved to. Currently only the "files" method is defined.

Example Usage:

```
my @result = $postfix->resolve_alias($alias, 'files');
```

rebuild_alias_map()

This function can be used to rebuild alias map in hash file using postalias utility.

Example Usage:

```
$postfix->rebuild_alias_map();
```

get_transports()

This function can be used to make a hash using /etc/postfix/transport configuration file, which will contain domain and destination, respectively as key and value pair. Hash reference will be sent as a result.

Example Usage:

```
my $transports = $postfix->get_transports();
```

add_transport(\$\$)

This function is used to add an entry in /etc/postfix/transport configuration file. It receives domain name and destination as arguments.

Example Usage:

```
$postfix->add_transport($domain_name, $dest);
```


update_transport()

This function is used to update an entry in /etc/postfix/transport configuration file. It receives domain name and destination as arguments.

Example Usage:

```
$postfix->update_transport($domain_name, $dest);
```

delete_transport(\$)

This function is used to delete an entry from /etc/postfix/transport configuration file. It receives domain name as argument. It also deletes related relaydomain too.

Example Usage:

```
$postfix->delete_transport($domain_name);
```

rebuild_transport_map()

This function can be used to rebuild transport map in hash file using postmap utility.

Example Usage:

```
$postfix->rebuild_transport_map();
```

3.9.5. AUTHORS

Pax Dickinson <pax@guardiandigital.com>

3.9.6. COPYRIGHT AND LICENSE

Copyright Guardian Digital, Inc., All Rights Reserved

3.10. reporting**3.10.1. NAME**

reporting - Guardian Digital WebTool module interface to system reports.

3.10.2. SYNOPSIS

```
use reporting;
my $reporting = new reporting($page);
```

3.10.3. DESCRIPTION

This Guardian Digital WebTool module provides an interface to listing and retrieving system reports.

3.10.4. METHODS

get_reptypes(\$)

This function returns array of report types available under /var/log/reports. It receives webtool UI object(page) as argument.

Example Usage:

```
my @reptypes = $reporting->get_reptypes($page);
```

get_repdates(\$)

This function returns array of report dates by reading /var/log/reports directory. If no dates found than by default it gives current localtime as report date. Array of dates in returned as result.

Example Usage:

```
my @reports = $reporting->get_repdates($reptype);
```

read_report(\$\$)

This function is used to read report file. It receives report name and date as argument and returns a report in array form as a result.

Example Usage:

```
@repdata = $reporting->read_report($reptype, $repdate);
```

3.10.5. AUTHORS

Pax Dickinson <pax@guardiandigital.com>

3.10.6. COPYRIGHT AND LICENSE

Copyright Guardian Digital, Inc., All Rights Reserved

3.11. services

3.11.1. NAME

services - Guardian Digital WebTool module interface to system services.

3.11.2. SYNOPSIS

```
use services;
my $services = new services();
```

3.11.3. DESCRIPTION

This Guardian Digital WebTool module provides an interface to manipulating system services. This module allows you to start and stop services, enable and disable services at boot time, and get the current and boot status for services. This function allows you to manipulate xinetd-based services.

3.11.4. METHODS

read_initscript(\$)

This function reads a given daemon file(usually under /etc/init.d/).It returns values of chkconfig and description in hash reference only if both exists in file. Chkconfig values consists of in which runlevel it should start, its start priority and stop priority.

Example Usage:

```
my $data = read_initscript(DIR_INITSRIPTS . '/' . $f);
next if (! defined($data) || $blacklist->{$data->{'name'}});
$data->{'current_state'} = $self->current_state($data->{'name'});
$data->{'boot_state'} = $self->boot_state($data->{'name'});
```

read_xinetd(\$)

This function reads a given daemon file(usually under /etc/xinetd.d/). It returns values of interface(what IPs it should listen to), service enable/disable and description in hash reference.

Example Usage:

```
my $tmp = read_xinetd(DIR_XINETD . '/' . $f);
$description = $tmp->{'description'};
$interface = $tmp->{'interface'};
$disable_status = $tmp->{'disable'};
```

boot_state(\$)

This function basically checks particular service is scheduled to start at booting or not. It takes service name as argument. It reads through applying chkconfig command and returns 1 if service is running or run level 3 is on.

Example Usage:

```
$boot_schedule = $services->boot_state($service);
```

current_state(\$)

This function can be used to check the current status of service by passing service name as argument. Returns 1 if service is running.

Example Usage:

```
$current_status = $services->current_state($service);
```

list_services()

This function basically lists all services from /etc/init.d/ and /etc/xinetd.d and return data like current state, boot state, name, description back in array form.

Example Usage:

```
foreach my $svc ($services->list_services()) {  
  my $service      = $svc->{'name'};  
  my $description  = $page->str_truncate(make_description($page, $svc), 60);  
  my $current_state = make_toggle_link($page, 'current', $svc->{'name'},  
                                       $svc->{'current_state'}, $svc->{'type'});  
  my $boot_state   = make_toggle_link($page, 'boot', $svc->{'name'},  
                                       $svc->{'boot_state'}, $svc->{'type'});  
}
```

run_initscript(\$\$)

Depending on the arguments (service name and operation), this function is used to start/stop/restart given service.(e.g /etc/init.d/named start). Returns true or false.

Example Usage:

```
$services->run_initscript($service, 'start');
```

start(\$)

This function is used to start a particular service, it receives a service name as an argument.

Example Usage:

```
if ($services->start('ftp') == 0) {
```

```
.....
}
```

stop(\$)

This function is used to stop a particular service, it receives a service name as an argument.

Example Usage:

```
if ($services->stop('ftp') == 0) {
.....
}
```

restart(\$)

This function is used to restart a particular service, it receives a service name as an argument.

Example Usage:

```
$services->restart('ftp');
```

run_chkconfig(\$\$)

This function is used to activate/deactivate service at boot time. It receives two arguments(operation=add/delete and service=service name). Its been done using /sbin/chkconfig --operation servicename command.

Example Usage:

```
$services->run_chkconfig('add', $service);
```

enable(\$)

This function is used to add(enable) service at boot time. It receives service name as argument.

Example Usage:

```
if ($services->enable('ftp') == 0) {
.....
}
```

disable(\$)

This function is used to delete(disable) service at boot time. It receives service name as argument.

Example Usage:

```
if ($services->disable('ftp') == 0) {
```

```
.....  
}
```

xinetd_enable(\$)

This function is used to enable inetd-based services(e.g.ftp,simap). it receives service name as an argument and opens service file under /etc/xinetd.d. In that file it comment out disable option. Then it restarts xinetd.

Example Usage:

```
$services->xinetd_enable('ftp');
```

xinetd_disable(\$)

This function is used to disable inetd-based services(e.g.ftp,simap). it receives service name as an argument and opens service file under /etc/xinetd.d. In that file it turn on disable option. Then it restarts xinetd.

Example Usage:

```
$services->xinetd_disable('ftp');
```

3.11.5. AUTHORS

Ryan W. Maple <ryan@guardiandigital.com>

3.11.6. COPYRIGHT AND LICENSE

Copyright Guardian Digital, Inc., All Rights Reserved

3.12. ssh

3.12.1. NAME

ssh - Guardian Digital WebTool module interface to the Secure Shell (SSH) service.

3.12.2. SYNOPSIS

```
use ssh;  
my $ssh = new ssh($page);
```

3.12.3. DESCRIPTION

This Guardian Digital WebTool module provides an interface to manipulating the Secure Shell service. This module allows you to manage the sshd_config configuration file and to generate SSH keys.

3.12.4. METHODS

config_read()

This function reads sshd-server systemwide configuration file (/etc/ssh/sshd_config), and returns hash reference containing options and their respective values as key-value pairs.

Example Usage:

```
$ssh_opts = $ssh->config_read();
```

config_write(\$)

This function updates /etc/ssh/sshd_config file, it receives a hash reference containing keys and values(set of directives to update).

Example Usage:

```
$ssh->config_write($newopts);
```

generate_key(\$)

This function is used to generate ssh keys, it creates .ssh directory under root if its not there, remove old keys if any, generates keys, put in appropriate folder and sets permission. It receives hash reference as argument containing information like keyfilename, passphrase, description etc.

Example Usage:

```
my $userinfo = {
    'keyfilename' => $keyfilename,
    'description' => $description,
    'passphrase'  => $passphrase,
}
my $keyfile = $ssh->generate_key($userinfo);
```

restart()

This function is used to restart the sshd service.

Example Usage:

```
$services->restart('sshd');
```

3.12.5. AUTHORS

Ryan W. Maple <ryan@guardiandigital.com>

3.12.6. COPYRIGHT AND LICENSE

Copyright Guardian Digital, Inc., All Rights Reserved

3.13. sysstat

3.13.1. NAME

sysstat - Guardian Digital WebTool module interface to system information and statistics.

3.13.2. SYNOPSIS

```
use sysstat;  
my $sysstat = new sysstat();
```

3.13.3. DESCRIPTION

This Guardian Digital WebTool module provides an interface for retrieving general system information and statistics. This module allows you to gather information such as memory, swap, network, and disk usage, RAID status (if available), SELinux status, process counts and lists, CPU temperature (if available), and system uptime.

This module is named "sysstat" instead of "index" because index is a perl reserved word.

3.13.4. METHODS

memstats()

This function should never be called directly. It is useful to get current memory statistics from /proc/memstats. It returns hash reference containing all parameters as key-value pairs.

Example Usage:

```
$memref = memstats();
```

load_avg()

This function should never be called directly. It returns 3 member array of load average numbers from /proc/loadavg.

Example Usage:

```
@loadavg = load_avg();
```


diskstats()

This function should never be called directly. It is used to get disk statistics in human readable form(/bin/df). Result parameters are returned in hash reference. see example for parameters(hash) details.

Example Usage:

```
my $diskref = diskstats();

$fs           = $diskref->{$filesystem}->'filesystem';
$fstype       = $diskref->{$filesystem}->'type';
$fssize      = $diskref->{$filesystem}->'size';
$fsused      = $diskref->{$filesystem}->'used';
$fsavail     = $diskref->{$filesystem}->'avail';
$fsusedpercent = $diskref->{$filesystem}->'usedpercent';
$fsmountpoint = $diskref->{$filesystem}->'mountpoint';
```

interfaces_list()

This function should never be called directly. It is used to get interfaces details from /etc/network/interfaces file. Depending on the interface type(lo,eth0,eth1 etc) it parses information like interface, protocol, type, virtual parent, address, net mask etc. Hash reference is return as result.

Example Usage:

```
$interfaces = interfaces_list();
```

count_processes()

This function returns a count for number of processes running (/proc).

Example Usage:

```
$count = $sysstat->count_processes();
```

uptime()

This function returns current system uptime(/proc/stat).

Example Usage:

```
$uptime = $sysstat->uptime();
```

print_load_avg()

This function returns 3 member array of load avg numbers as comma separated values.

Example Usage:

```
$comma_separated = $sysstat->print_load_avg();
```

mem_usage()

This function calculates and returns percentage(%) of memory used. It also shows memory used in MB out of total MB.

Example Usage:

```
$mempercent = $sysstat->mem_usage();
```

swap_usage()

This function calculates and returns percentage(%) of swap memory used. It also shows memory used in MB out of total MB.

Example Usage:

```
$swappercent = sysstat->swap_usage();
```

disk_usage()

This function returns disk usage(size, used, available, used percent, mount point) in table format.

Example Usage:

```
$disk_usage_table = sysstat->disk_usage();
```

network_usage()

This function returns network usage(interface statistics: link, address, currentin, currentout, total) in table format.

Example Usage:

```
$network_usage_table = $sysstat->network_usage();
```

read_mrtg(\$)

This function retrieves the daily, weekly, monthly and yearly MRTG data for the given argument (cpu, ram etc). Hash reference is returned as a result.

Example Usage:

```
$usage = $sysstat->read_mrtg("cpu");
```

hostname()

This function is used to get hostname of the system.

Example Usage:

```
$name_of_host = $sysstat->hostname();
```

raid_status()

This function is used to get status of RAID. Currently it returns value "Optimal" by default.

Example Usage:

```
$r_status = $sysstat->raid_status();
```

shell_users()

This function is used to get list of current shell users. It returns hash reference containing information like tty,user,date and host. see example.

Example Usage:

```
my $users = $sysstat->shell_users();
$user = $users->{'tty0'}->{user};
$date = $users->{'tty0'}->{date};
$host = $users->{'tty0'}->{host};
```

selinux_status()

This function is used to get current status of SELinux. First it checks if its disabled entirely or not, and then it returns current mode(permissive or enforcing).

Example Usage:

```
$current_status = $sysstat->selinux_status();
```

kernel_ver()

This function is used to get version of kernel.

Example Usage:

```
$kernel_version = $sysstat->kernel_ver();
```

engarde_ver()

This function is used to get version of engarde linux.

Example Usage:

```
$engarde_version = $sysstat->engarde_ver();
```

cputemp()

This function is used to get cpu's current temperature in ferenheit.

Example Usage:

```
my $temp = $sysstat->cputemp();
```

list_processes(\$)

This function can be used to list current processes running in desired order. sort order(mem,cpu or user) is passed as an argument. All information is returned in array of hash references.

Example Usage:

```
@proc = $sysstat->list_processes('cpu');
```

dmesg()

This function is used to get last 10 lines(default) from the system log in array format.

Example Usage:

```
@loglines = $sysstat->dmesg();
```

webtool_audit()

This function is used to get last 10 lines(default) from the webtool audit log in array format.

Example Usage:

```
@webtoollog = $sysstat->webtool_audit();
```

update_news()

This function is used to update gdsn news file if its older than 12 hours. If error occurs during this operation then it returns to the calling function.

Example Usage:

```
$err = $sysstat->update_news();
```

3.13.5. AUTHORS

Pax Dickinson <pax@guardiandigital.com>

3.13.6. COPYRIGHT AND LICENSE

Copyright Guardian Digital, Inc., All Rights Reserved

3.14. users

3.14.1. NAME

users - Guardian Digital WebTool module interface to system users and groups.

3.14.2. SYNOPSIS

```
use users;
my $users = new users;
```

3.14.3. DESCRIPTION

This Guardian Digital WebTool module provides an interface to manipulating system users and groups. This module allows you to manage system users, system groups, and primary and secondary groups for users.

3.14.4. METHODS

map_grent_array(\$)

This function is used internally, it receives array as returned by `getgrent()`. It just makes a hash reference containing this information and returns it.

Example Usage:

```
my $hashgrent = users::map_grent_array(\@u);
```

map_pwent_array(\$)

This function receives array as returned by `getpwent()`. It sets all those parameters as key-value pairs including shell and type of users and returns hash reference.

Example Usage:

```
my $hashpwent = map_pwent_array(\@u);
```

set_password(\$\$)

This function is basically used to set password for particular user. Username and password is passed as arguments.

Example Usage:

```
users::set_password($username, $password);
```

update_group_members(\$\$)

This function is basically used to update the group members. Groupname and members are sent as arguments to this function.

Example Usage:

```
users::update_group_members($name, $members);
```

users_list()

This function returns all of the local users on the system excluding root.

Example Usage:

```
my @tmp = $users->users_list();
```

user_get(\$@)

This function is used to get a particular user related information(returned by getpwent). Username is passed as an argument.

Example Usage:

```
my $user = $users->user_get($username);
```

user_create(\$)

This function is used to create a user in the system, It receives a hash reference containing user parameters like username, login shell, home directory, group and password.

Example Usage:

```
my $u = {  
  'username' => $in{'username'},  
  'gcos' => $in{'name'},  
  'group' => make_group($in{'type'}, $in{'group'}),  
  'password' => $in{'password'},
```

```

        'shell'          => make_shell($in{'shell_axs'}),
    };

    $users->user_create($u);

```

user_modify(\$)

This function is used to modify a current user, It receives hash reference as argument which contains parameters to be modified (any parameters listed in user_create function).

Example Usage:

```
$users->user_modify($u);
```

user_delete(\$@)

This function is used to delete user's account from system. It also receives one optional argument besides username, depends on that user's home directory deletion can be decided. If that parameter is sent other than zero than that user's home directory will be deleted.

Example Usage:

```
$users->user_delete($username, $killhome);
```

groups_list(\$)

This function is basically used to retrieve array references as returned by getgrnt of all users falling under groups (root/admin,users and group id is greater than 500) and returns it in array form.

Example Usage:

```
my @grouplist = $users->groups_list();
```

group_get(\$@)

This function can be used to receive arrays as returned by getgrnam. It receives group name as mandatory argument. Hash reference containing key-value pairs of parameters and values(from getgrnam) is returned.

Example Usage:

```
my $groupdetail = $groups->group_get($group);
```

group_create(\$)

This function is used to create a new group. It receives hash reference containing new group name and list of members belongs to that group.

Example Usage:

```
my $group = {  
    'name'      => $in{'name'},  
    'members'  => join(',', $in{'members'}),  
};  
  
$users->group_create($group);
```

group_modify(\$)

This function is used to modify an existing group(e.g new name). It receives hash reference as argument which contains parameters to be modified (any parameters listed in group_create function).

Example Usage:

```
$users->group_modify($group);
```

group_delete(\$)

This function is used to delete an existing group. It receives a group name as an argument.

Example Usage:

```
$users->group_delete($group_name);
```

3.14.5. AUTHORS

Ryan W. Maple <ryan@guardiandigital.com>

3.14.6. COPYRIGHT AND LICENSE

Copyright Guardian Digital, Inc., All Rights Reserved

3.15. webtool

3.15.1. NAME

webtool - Guardian Digital WebTool module interface to... itself!

3.15.2. SYNOPSIS

```
use webtool;  
my $webtool = new webtool;
```


3.15.3. DESCRIPTION

This Guardian Digital WebTool module provides an interface to manipulating the Guardian Digital WebTool. This module allows you to manage WebTool users, their language, and their access rights.

3.15.4. METHODS

encrypt_password(\$)

This function is used to encrypt a password string. It receives password string as an argument, it returns encrypted password.

Example Usage:

```
$encrypted_password = $webtool->encrypt_password($password);
```

enumerate_languages()

This function returns an array of the translations currently defined in the WebTool, suitable for use in a `$page->input_select()`.

Example Usage:

```
my @languages = $webtool->enumerate_languages();
```

read_acl_files()

Essentially this function reads `.acl` files located under each module directory and returns a hash reference containing values of major, minor and description for each file under that directory.

Example Usage:

```
my $acls = $webtool->read_acl_files();
```

read_acl_major_minor(\$)

Essentially this function reads values of major, minor and description, It receives file(.acl) as argument and returns result in array form.

Example Usage:

```
my $tmp = $webtool->read_acl_major_minor($acl);
```

restart()

This function is used to restart the webtool service.

Example Usage:

```
$webtool->restart ();
```

account_create(\$)

This function is used to create/update webtool account. It receives hash reference as account details which contains username, language, password etc.

Example Usage:

```
my $a = {  
    'username'    => $in{'username'},  
    'password'    => $in{'password1'},  
    'language'    => $in{'language'},  
};  
  
$webtool->account_create($a);
```

account_delete(\$)

This function is used to delete webtool account. It receives webtool accountname as argument.

Example Usage:

```
$webtool->account_delete($accountname);
```

account_get(\$)

This function is used to get information of webtool account. It receives username(webtool account name) as an argument. It returns account detail in hash reference(username, password, language).

Example Usage:

```
my $accountdetail = $webtool->account_get($accountname);
```

list_accounts()

This function is used to get a list of all listed users in /etc/webtool/miniserv.users file and returns details(username, password, language) in hash reference.

Example Usage:

```
my $accounts = $self->list_accounts();  
foreach my $u (sort keys %{$accounts}) {  
    my $user = $accounts->{$u}->{'username'};
```

```
my $pass    = $accounts->{$u}->{'password'};  
my $lang    = $accounts->{$u}->{'language'};  
}
```

3.15.5. AUTHORS

Ryan W. Maple <ryan@guardiandigital.com>

3.15.6. COPYRIGHT AND LICENSE

Copyright Guardian Digital, Inc., All Rights Reserved

