

# Wanna be a BlueQuartz developer?

Takashi Matsuo <tmatsuo@10art-ni.co.jp>

For BlueQuartz Conference 2004.

# Where is our destination?

- ◆ We're gonna try to use sausalito-devel-tools.
  - We will rpmize our first application.
- ◆ We're gonna try to use CCE(Cobalt Configuration Engine).
  - We will define a CODB object, and create one.
  - We will define a CCE handler, to store data to a system file.

# What's sausalito-devel-tools?

```
$ rpm -ql sausalito-devel-tools
/usr/sausalito/bin/AutoBuild.pl
/usr/sausalito/bin/buildFromBTO.pl
/usr/sausalito/bin/checkout_product.pl
/usr/sausalito/bin/cpan2rpm
/usr/sausalito/bin/cvs_checkout
/usr/sausalito/bin/cvs_checkout_multibranch
/usr/sausalito/bin/dumpCaps
/usr/sausalito/bin/ftp_put
/usr/sausalito/bin/hup
/usr/sausalito/bin/makePackageVars
/usr/sausalito/bin/makePkg
/usr/sausalito/bin/make_install
/usr/sausalito/bin/make_release
/usr/sausalito/bin/make_release_spec
/usr/sausalito/bin/mendocino_package
/usr/sausalito/bin/mod_rpmize
/usr/sausalito/bin/po_automerge.pl
/usr/sausalito/bin/reencode.pl
/usr/sausalito/bin/verifyRelease.pl
/usr/sausalito/devel/defines.mk
/usr/sausalito/devel/libs.mk
/usr/sausalito/devel/module.mk
/usr/sausalito/devel/package.mk
/usr/sausalito/devel/rules.mk
/usr/sausalito/devel/templates
/usr/sausalito/devel/templates/rpmdefs.tmpl
/usr/sausalito/devel/templates/spec.tmpl
/usr/sausalito/include/libdebug.h
/usr/sausalito/lib/libdebug.a
/usr/sausalito/perl/BTO.pm
/usr/sausalito/perl/Build.pm
/usr/sausalito/perl/Devel.pm
```

It is tools for development. That's it.

We can use

`/usr/sausalito/devel/defines.mk`

`/usr/sausalito/devel/module.mk`

to rpmize our applications.

I made a patch against

`sausalito-devel-tools` which enables us to  
'make rpm' as non root user.

# Development environment

- ◆ Fedra Core 1 reference kit + BlueQuartz-5100R-20040620. This is the base system.
- ◆ Install the following kinds of applications using yum.  
cvs rpm-build gettext gcc ctags tree
- ◆ Get the sources of BlueQuartz via cvs.
- ◆ Get the following from the FTP site of BlueQuartz and install them.  
glib-devel-1.2.10-11BQ2.i386.rpm  
glib-ghash-1.0-1BQ1.i386.rpm
- ◆ Apply my patch against sausalito-devel-tools, then re-rpmize it, and install new rpm.
- ◆ Prepare for rpmbuild.
  - Put following line to ~/.rpmmacros.  
%\_topdir /home/.users/42/tmatsuo/rpm
  - Make directories.  
mkdir -p ~/rpm/{BUILD,SOURCES,SPECS,SRPMS,RPMS}

# Hello world application

hello.html

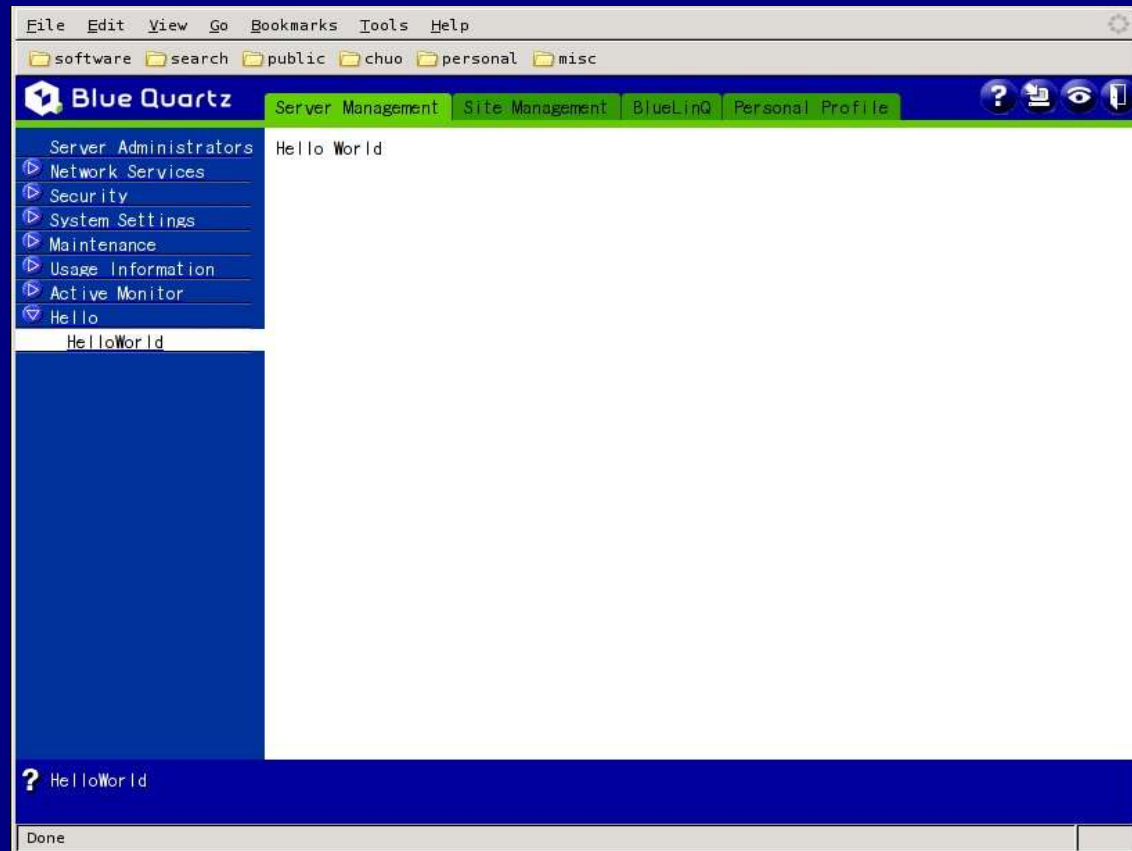
```
<html>
<body>
Hello World
</body>
</html>
```

hello.xml

```
<item
  id="base_hello"
  label="Hello"
  description="Hello"
  requiresChildren="1">
  <parent id="base_sysmanage" order="90"/>
</item>
```

helloworld.xml

```
<item
  id="base_helloWorld"
  label="HelloWorld"
  description="HelloWorld"
  url="/base/hello-simple/hello.html">
  <parent id="base_hello" order="10"/>
</item>
```



# Directory structure

Explore directory structure of the other module.

```
$ cd src/ui/base-user.mod
$ ls
CVS          VERSION      glue         packing_list  templates
Makefile     constructor  locale       perl          ui
$ tree -l CVS ui
ui
|-- menu
|   |-- personalAccount.xml
|   |-- personalEmail.xml
|   |-- userList.xml
|-- web
|   |-- emailAliasGenerator.JS
|   |-- personalAccount.php
|   |-- personalAccountHandler.php
|   |-- personalEmail.php
|   |-- personalEmailHandler.php
|   |-- user.inc
|   |-- userAdd.php
|   |-- userAddHandler.php
|   |-- userDefaults.php
|   |-- userDefaultsHandler.php
|   |-- userList.php
|   |-- userMod.php
|   |-- userModHandler.php
|   |-- userNameGenerator.JS
|   |-- userRemoveHandler.php
```

2 directories, 18 files

There are xml files in 'ui/menu' and php files in 'ui/web'.

# Make directories

Make directories for 'Hello World application' and copy files from base-user.mod.

```
$ cd ~/src/ui
$ mkdir base-hello-simple.mod
$ cd base-hello-simple.mod
$ cp -r ../base-user.mod/{Makefile,VERSION,packing_list,templates} .
```

## Edit Makefile.

```
$ vi Makefile
(snip..)
SERVICE=hello-simple
```

```
VERSION=1.1.0
RELEASE=111BQ1
```

```
# add a buildarch if desired
BUILDARCH=noarch
```

```
# locale exclude pattern
#XLOCALEPAT=
```

```
BUILDUI=yes
BUILDGLUE=no
BUILDLOCALE=no
BUILDSRC=no
```

```
#####
# some useful defines
INSTALL=install
INSTALL_BIN=$(INSTALL) -m 755
INSTALL_OTH=$(INSTALL) -m 644
TMPDIR=/tmp
```

```
include /usr/sausalito/development/module.mk
```

# Edit files

Edit files in the directory 'templates'.

```
$ vi templates/rpmdefs.tmpl  
(snip..)  
<begin locale>  
[AUTOFILL]
```

```
<end locale>
```

```
<begin glue>  
/etc/skel/user/en/web/  
/etc/skel/user/ja/web/  
/etc/skel/user/es/web/  
/etc/skel/user/fr/web/  
/etc/skel/user/de/web/  
[AUTOFILL]  
<end glue>  
(snip..)
```

```
$ vi templates/spec.tmpl  
(snip..)  
%changelog  
* Wed Aug 25 2004 Takashi Matsuo <tmatsuo@10art-ni.co.jp>  
- new, Hello world application.  
  
* Tue Apr 18 2000 Adrian Sun <asun@cobalt.com>  
- new, improved spec file template.  
  
* Thu Mar 30 2000 Adrian Sun <asun@cobalt.com>  
- sample spec file for skeleton package.
```

# Put the essential files

```
$ mkdir -p ui/{menu,web}
```

```
$ vi ui/web/hello.html
```

```
<html>  
<body>  
Hello World  
</body>  
</html>
```

```
$ vi ui/menu/hello.xml
```

```
<item id="base_hello" label="Hello" description="Hello" requiresChildren="1">  
  <parent id="base_sysmanage" order="90"/>  
</item>
```

```
$ vi ui/menu/helloworld.xml
```

```
<item id="base_helloWorld" label="HelloWorld" description="HelloWorld" url="/base/hello-simple/hello.html">  
  <parent id="base_hello" order="10"/>  
</item>
```

```
$ tree ui
```

```
ui  
|-- menu  
|   |-- hello.xml  
|   |-- helloworld.xml  
|-- web  
|   |-- hello.html
```

```
2 directories, 3 files
```

# Make rpms

```
$ make rpm  
(snip..)
```

```
$ ls as_rpms
```

```
base-hello-simple-capstone-1.1.0-111BQ1.noarch.rpm
```

```
base-hello-simple-ui-1.1.0-111BQ1.noarch.rpm
```

```
$ sudo rpm -ivh as_rpms/*.rpm
```

```
Preparing... ##### [100%]  
 1:base-hello-simple-ui ##### [ 50%]  
 2:base-hello-simple-capst##### [100%]
```

Success!!

Can you see it on your browser?

# Greetings application

- ◆ Define a CODB object named 'Greeting'.
- ◆ Listing, adding your object from browsers.
- ◆ Register handlers to your object, to update system file when an object is created or destroyed.

# Preparation-1

We start it from a copy of 'Hello World application'.(vague)

```
$ cp -r base-hello-simple.mod base-greetings.mod
```

## Edit Makefile

```
# specify the following variables:
```

```
VENDOR=base
```

```
# if the VENDOR field is an alias, this should be the real name.
```

```
# otherwise, set it to VENDOR.
```

```
VENDORNAME=cobalt
```

```
SERVICE=greetings
```

```
VERSION=1.1.0
```

```
RELEASE=111BQ1
```

```
# add a buildarch if desired
```

```
BUILDARCH=noarch
```

```
# locale exclude pattern
```

```
#XLOCALEPAT=
```

```
BUILDUI=yes
```

```
BUILDGLUE=no
```

```
BUILDLOCALE=no
```

```
BUILDSRC=no
```

```
#####
```

```
# some useful defines
```

```
INSTALL=install
```

```
INSTALL_BIN=$(INSTALL) -m 755
```

```
INSTALL_OTH=$(INSTALL) -m 644
```

```
TMPDIR=/tmp
```

```
include /usr/sausalito/devel/module.mk
```

# Preparation-2

Rename files under ui directory.

```
$ mv ui/web/hello.html ui/web/greetings.php
$$ mv ui/menu/hello.xml ui/menu/greetings.xml
$ mv ui/menu/helloworld.xml ui/menu/viewGreetings.xml
```

Edit greetings.xml.

```
$ vi ui/menu/greetings.xml
<item id="base_greetings" label="greetings" description="greetings" requiresChildren="1">
  <parent id="base_sysmanage" order="95"/>
</item>
```

Edit viewGreetings.xml.

```
$ vi ui/menu/viewGreetings.xml
<item id="base_greetings_view" label="greeting list" description="greetings" url="/base/greetings/greetings.php">
  <parent id="base_greetings" order="10"/>
</item>
```

Then 'make rpm' and install generated rpms.  
Can you see 'greetings' menu on the left side?

# Define Greeting object 1

Make directories for our object.

```
$ mkdir -p glue/{conf,handlers,schemas}
```

Write a schema file of Greeting object. This object is very simple, and has only one property.

```
$ vi glue/schemas/greeting.schema
```

```
<!--
$id:$

This is the schema for greeting object.
-->
<typedef name="words" type="re"
  data = "^[:\n\r]*$"
/>
<class name="Greeting" namespace="" version="1.0">
  <property name="words" type="words"/>
</class>
```

Turn 'BUILDGLUE=no' into 'BUILDGLUE=yes' in Makefile, and make rpm.  
You will get base-greetings-glue-1.1.0-111BQ1.noarch.rpm.

```
$ ls as_rpms
base-greetings-capstone-1.1.0-111BQ1.noarch.rpm
base-greetings-glue-1.1.0-111BQ1.noarch.rpm
base-greetings-ui-1.1.0-111BQ1.noarch.rpm

$ rpm -qpl as_rpms/base-greetings-glue-1.1.0-111BQ1.noarch.rpm
/usr/sausalito/schemas/base/greetings
/usr/sausalito/schemas/base/greetings/greeting.schema
```

# Define Greeting object 2

Install new rpm, and try to create our object using cceclient.

```
$ sudo rpm -ivh as_rpms/base-greetings-glue-1.1.0-111BQ1.noarch.rpm
Preparing...                               ##### [100%]
 1:base-greetings-glue                     ##### [100%]
```

```
$ /usr/sausalito/bin/cceclient
100 CSCP/0.80
200 READY
auth admin *****
109 SESSIONID Jc6JF5musm2b7p47EwzbuWCXqfSH46HB6DbCyIWPYnQt2Kow4LvovY9L1PgVLNI
201 OK
find Greeting
301 UNKNOWN CLASS Greeting
401 FAIL
bye
202 GOODBYE
```

'UNKNOWN CLASS Greeting'. Do we have to restart CCED?

```
$ sudo /etc/init.d/cced.init reload
Reloading cced configuration: done
$ /usr/sausalito/bin/cceclient
100 CSCP/0.80
200 READY
auth admin *****
109 SESSIONID vJ6EAyaOHjcVHyhVQMDMQXjbnxHsudkNKeh8ChN9pNTWb0FPA6rgUyg7WNqe0yR
201 OK
find Greeting
201 OK
```

It looks fine.

# Define Greeting object 3

Create Greeting object.

```
create Greeting words="Hello world"  
104 OBJECT 42  
201 OK
```

Greate!!

```
find Greeting  
104 OBJECT 42  
201 OK  
get 42  
102 DATA NAMESPACE = ""  
102 DATA words = "Hello world"  
102 DATA CLASSVER = "1.0"  
102 DATA CLASS = "Greeting"  
102 DATA OID = "42"  
201 OK  
bye  
202 GOODBYE
```

Success!!

# List our object on UI.

Edit ui/web/greeting.php to list objects on browser.

```
$ vi ui/web/greetings.php
<?php
    include_once("ServerScriptHelper.php");
    $helper =& new ServerScriptHelper();
    $factory =& $helper->getHtmlComponentFactory("base-greetings");
    $cce =& $helper->getCceClient();
    $page =& $factory->getPage();
    $greetingList = $factory->getScrollList("greetings",
        array("words"), array()
    );
    print $page->toHeaderHtml();
    $greetings = $cce->findx('Greeting', array(), array(), 'locale', 'words');

    $greetingList->setColumnWidths(array("100%"));

    for ($i = 0; $i < count($greetings); $i++) {
        $greeting = $cce->get($greetings[$i]);
        $words = $factory->getSimpleText($greeting['words']);
        $greetingList->addEntry(array($words), "", false, $i);
    }
    print $greetingList->toHtml();
    print $page->toFooterHtml();
    $helper->destructor();
?>
```

I'll show you a screenshot on the next page.

File Edit View Go Bookmarks Tools Help

software search public chuo personal misc

# Blue Quartz

Server Management Site Management BlueLinQ Personal Profile

- Server Administrators
- Network Services
- Security
- System Settings
- Maintenance
- Usage Information
- Active Monitor
- Hello
- greetings

greeting list

greetings	
	1 Entry
words	
Hello world	

? greetings

Done

# Save our object 1

We need 'Add' button to add objects from browsers.

```
$ vi ui/web/greetings.php
<?php
include_once("ServerScriptHelper.php");
$helper =& new ServerScriptHelper();
$factory =& $helper->getHtmlComponentFactory("base-greetings");
$cce =& $helper->getCceClient();
$page =& $factory->getPage();
$greetingList = $factory->getScrollList("greetings",
    array("words"), array()
);
print $page->toHeaderHtml();
$greetings = $cce->findx('Greeting', array(), array(), 'locale', 'words');

$greetingList->setColumnWidths(array("100%"));

$greetingList->addButton(
    $factory->getAddButton
    ('/base/greetings/addGreeting.php', 'greeting_help'));

for ($i = 0; $i < count($greetings); $i++) {
    $greeting = $cce->get($greetings[$i]);
    $words = $factory->getSimpleText($greeting['words']);
    $greetingList->addEntry(array($words), "", false, $i);
}
print $greetingList->toHtml();
print $page->toFooterHtml();
$helper->destructor();
?>
```



# Save our object 2

We need a form to add objects.

```
$ vi ui/web/addGreeting.php
<?php
    include_once("ServerScriptHelper.php");
    $helper =& new ServerScriptHelper();
    $factory =& $helper->getHtmlComponentFactory("base-greetings",
"/base/greetings/addGreetingHandler.php");
    $cce =& $helper->getCceClient();
    $page =& $factory->getPage();
    $form = $page->getForm();
    $formId = $form->getId();

    $block = $factory->getPagedBlock("addGreeting");
    $block->addFormField(
        $factory->getTextField("words", $words, "rw"),
        $factory->getLabel("words")
    );
    $block->addButton($factory->getSaveButton($page->getSubmitAction()));
    $block->addButton($factory->getCancelButton("/base/greetings/greetings.php"));print $page->toHeaderHtml();
    print $block->toHtml();
    print $page->toFooterHtml();
    $helper->destructor();
?>
```



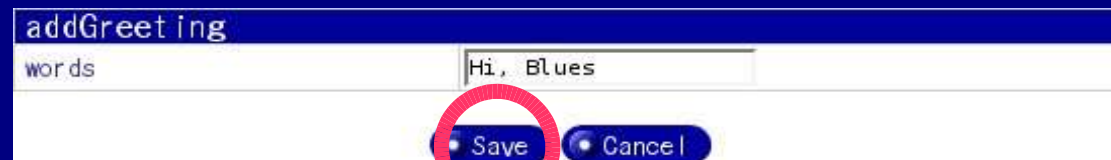
The screenshot shows a web browser window with a title bar that says "addGreeting". Below the title bar is a form with a single text input field. The label "words" is positioned to the left of the input field. At the bottom of the form, there are two buttons: "Save" and "Cancel".

# Save our object 3

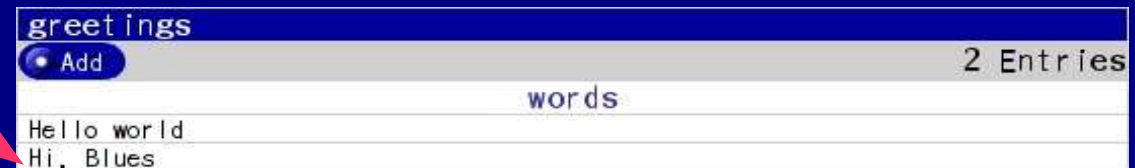
Finally, we need handler php which can process attributes posted from the form.

```
$ vi ui/web/addGreetingHandler.php
<?php
include_once("ServerScriptHelper.php");
$helper = new ServerScriptHelper();
$cce = $helper->getCceClient();
$errors = array();
$attr = array("words" => $words);
$oid = $cce->create("Greeting", $attr);
$errors = array_merge($errors, $cce->errors());
print(
    $helper->toHandlerHtml("/base/greetings/greetings.php", $errors, false)
);
$helper->destructor();
?>
```

Now, we can add object from browsers.



addGreeting	
words	Hi, Blues
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	



greetings		2 Entries
<input type="button" value="Add"/>	words	
	Hello world	
	Hi, Blues	

# Handler registration 1

Edit glue/conf/greeting.conf to register handlers.

```
$ vi glue/conf/greeting.conf
Greeting._CREATE perl:base/greetings/handle_greeting.pl
Greeting._DESTROY perl:base/greetings/delete_greeting.pl
```

Write a handler script which is activated when an object is created.(Long, extend into next page)

```
$ vi glue/handlers/handle_greeting.pl
#!/usr/bin/perl -w -l/usr/sausalito/perl
use CCE;
```

```
my $cce = new CCE;
$cce->connectfd();
```

```
my $config_file = '/etc/greeting.conf';
```

```
my $oid = $cce->event_oid();
my $old = $cce->event_old();
my $new = $cce->event_new();
my $obj = $cce->event_object();
my ($sysoid) = $cce->find("System");
my ($sysobj);
{
    my $ok;
    ($ok, $sysobj) = $cce->get($sysoid);
}
```

```
if (conf_has_words($new->{'words'})) {
    fail();
} else {
    write_words($new->{'words'});
}
```

```
$cce->bye('SUCCESS');
exit(0);
```

# Handler registration 2

```
sub fail() {
    $cce->bye('FAIL');
    exit(1);
}

sub conf_has_words($) {
    my $words = shift;
    open (FILE, "$config_file") || return;
    my $ret = 0;
    while (<FILE>) {
        chomp;
        if ($_ =~ /^$words$/) {
            $ret = 1;
        }
    }
    close(FILE);
    return $ret;
}

sub write_words($) {
    my $words = shift;
    open (FILE, ">>$config_file") || fail();
    flock(FILE, 2);
    seek(FILE, 0, 2);
    print FILE "$words\n";
    close(FILE);
}
```

# Handler registration 3

Write a handler script which is activated when an object is destroyed.

```
$ vi glue/handlers/delete_greeting.pl
#!/usr/bin/perl -w -I/usr/sausalito/perl
use CCE;
```

```
my $cce = new CCE;
$cce->connectfd();
my $conf_file = '/etc/greeting.conf';
```

```
my $oid = $cce->event_oid();
my $old = $cce->event_old();
```

```
my $all_words = "";
my $words = $old->{'words'};
```

```
unless (-f $conf_file) {
    open(FILE, ">$conf_file") || fail();
} else {
    open(FILE, "+<$conf_file") || fail();
}
flock(FILE, 2);
seek(FILE, 0, 0);
while (<FILE>) {
    chomp;
    unless ($_ =~ /^$words$/) {
        $all_words .= "$_\n";
    }
}
truncate(FILE, 0);
seek(FILE, 0, 0);
print FILE $all_words;
close (FILE);
```

```
$cce->bye('SUCCESS');
exit(0);
```

```
sub fail {
    $cce->bye('FAIL');
    exit(1);
}
```

# Handler registration 4

Make rpm, install rpms, reload cced.  
Destroy objects from command line.

```
$ /usr/sausalito/bin/cceclient
100 CSCP/0.80
200 READY
auth admin *****
109 SESSIONID EnFdfLi3P0NzzDfGdMMzRE5sEyb9yat0nW3rx9iaZWympDTrgvOVZldr6dotdHj
201 OK
find Greeting
104 OBJECT 42
104 OBJECT 43
201 OK
destroy 42
201 OK
destroy 43
201 OK
bye
202 GOODBYE
```

Now, there is no Greeting object. You can see it on your browser.



/etc/greeting.conf is empty.

```
$ sudo cat /etc/greeting.conf
$
```

# Handler registration 5

Add object again.

addGreeting	
words	HelloWorld!!
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	

greetings	
<input type="button" value="Add"/>	1 Entry
words	
HelloWorld!!	

Is there the words in /etc/greeting.conf?

```
$ sudo cat /etc/greeting.conf  
HelloWorld!!
```

Good!!

