

BlueQuartz の開発者になりたい!!

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

For BlueQuartz Conference 2004.

BlueQuartz に関わるきっかけ 1

◆ Linux World Expo 2004 での 当社の発表

オープンソース導入の課題はこうして解決!
- テンアートニのオープンソースソリューション -

喜多 伸夫

株式会社テンアートニ

LinuxWorld Expo/Tokyo 2004 株式会社テンアートニ

本日のテーマ

- オープンソースがソフトウェア開発を変える!
 - オープンソース誕生の歴史的背景と特徴
- オープンソースと企業システム
 - アドバンテージと課題
 - 課題解決への道 - テンアートニのソリューション
 - オープンソースが Web アプリケーションを変える - オープンソース J2EE
- オープンソースプロジェクトへの取り組み

LinuxWorld Expo/Tokyo 2004 株式会社テンアートニ

BlueQuartz に関わるきっかけ 2

◆ Project BlueQuartz への協力を表明

Blue Quartz

Project Blue Quartz

- 2003 年 12 月、Sun Microsystems が Sun Cobalt RaQ 550 のソースコードを Open Source としてリリース
 - 2003 年 7 月の Qube3 のリリースに続くもの
- Project Blue Quartz とは、Cobalt Users Group がオープンソースの Cobalt ソフトウェアの他環境への移植作業をサポートする活動に付けられた名前



LinuxWorld Expo/Tokyo 2004 株式会社テンアートニ

Blue Quartz

- テンアートニは Project Blue Quartz の開発支援を通じ Cobalt Users Group の活動に貢献していく
- Qube3 の機能を現行の Blue Quartz Hosting Edition (550 base) への取り込みや、
 - ハードウェア検証などを検討中



LinuxWorld Expo/Tokyo 2004 株式会社テンアートニ

そんな訳で抜擢？されたのが私です

今回の目標

- ◆ sausalito-devel-tools を使ってみる。
 - 僕らの helloworld を rpm にする
- ◆ CCE を使ってみる。
 - CODB オブジェクトの定義・生成
 - CCE ハンドラの定義

sausalito-devel-tools とは

```
$ 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
```

名前の通りの開発支援ツール群

自分のアプリケーションを make rpm するには
/usr/sausalito/devel/defines.mk
/usr/sausalito/devel/module.mk
を間接的に使う事になる。

/usr/sausalito/devel/module.mk
では、root での作業が前提になっていたので
patch を作成して、一般ユーザで make rpm が
出来るようにした。

(ファイル名を見ると、他にも色々出来そう)

開発環境

- ◆ FC1 リファレンスキット + BlueQuartz-5100R-20040620 が基本
- ◆ yum で以下をインストール
cvs rpm-build gettext gcc ctags tree
- ◆ cvs で source 取得
- ◆ BlueQuartz の ftp サイトから、以下をダウンロードしてインストール
glib-devel-1.2.10-11BQ2.i386.rpm
glib-ghash-1.0-1BQ1.i386.rpm
- ◆ 一般ユーザで make rpm 出来るようにした patch を当てて、sausalito-devel-tools を作り直してインストール
- ◆ 一般ユーザ作成、rpmbuild の準備
 - ~/.rpmmacros に以下のように書く
%_topdir /home/.users/42/tmatsuo/rpm
 - ディレクトリ作成
mkdir -p ~/rpm/{BUILD,SOURCES,SPECS,SRPMS,RPMS}

hello world アプリケーション

hello.html

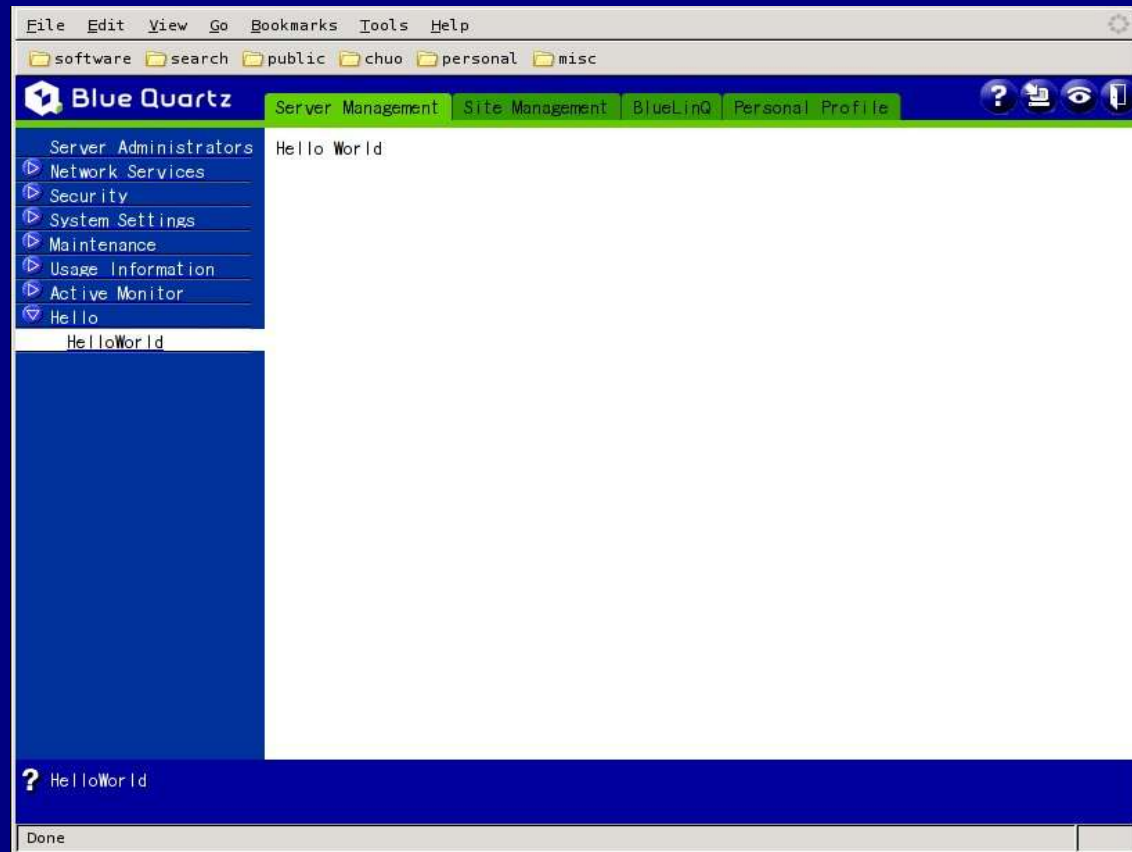
```
<html>
<body>
Hello World
</body>
</html>
```

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>
```



ディレクトリ構造

他のモジュールのディレクトリ構造を参考にしましょう。

```
$ 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

ui/menu には xml があって、ui/web に php がありますね。

ディレクトリの準備

Hello World のディレクトリを作り、base-user.mod からファイルを拝借して始めます。

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

Makefile を書き換えます。

```
$ 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/devel/module.mk
```

ファイルの編集

templates 以下のファイルを編集します。

```
$ vi templates/rpmdefs.tpl
(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.tpl
(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.
```

アプリケーション本体の配置

```
$ 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
```

rpm 作成、インストール

```
$ 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%]
```

成功しました！！ブラウザで動作確認してみてください！！
なお、この作業過程は `ttyplay` でもご覧いただけます。ご覧になりますか (y/n)?

Greetings アプリケーション

- ◆ CODB の Greeting オブジェクトの定義
- ◆ ui で 一覧・追加機能
- ◆ handler を定義して、オブジェクトの追加及び削除時に `/etc/greeting.conf` を更新する

モジュール作成の準備 1

以前作った HelloWorld アプリケーションのディレクトリをコピーして始めます。(いいかげんだなー)

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

Makefile を書き換えます

```
# 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=11BQ1
```

```
# 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
```

モジュール作成の準備 2

ui 以下のファイルはリネームします。

```
$ 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
```

greetings.xml を書き換えます。

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

viewGreetings.xml を書き換えます。

```
$ 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>
```

この段階で、make rpm 及び動作テストを試してみましょう。
Hello メニューの下に同じように greetings メニューが表示されれば成功です。

Greeting オブジェクトの定義 1

base-user.mod のディレクトリ構造をまねて glue ディレクトリツリーを作成します。

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

Greeting オブジェクトのスキーマ定義ファイルを書きます。プロパティをひとつしか持たないごく単純なオブジェクトです。

```
$ 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>
```

Makefile で、 BUILDGLUE=yes に変更し、 make rpm をすると先ほどの schema ファイルを含んだ base-greetings-glue-1.1.0-111BQ1.noarch.rpm が作成されます。

```
$ 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
```

Greeting オブジェクトの定義 2

schema をインストールして、コマンドラインの cceclient で接続し、テストします

```
$ 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
```

「Greeting なんてクラス知りません」と言われました。cced の設定を再読み込みしてもう一度。

```
$ 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
```

おおっ。良いみたいですね。

Greeting オブジェクトの定義3

では、Greeting オブジェクトを作ってみましょう。

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

素晴らしい。

```
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
```

思った通りに出来ました！！

オブジェクトを ui で表示する

コマンドラインで作ったオブジェクトを web の ui で表示しましょう。 ui/web/greeting.php を編集します。

```
$ 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();
?>
```

赤字の部分で、Greeting オブジェクトの words プロパティを取得して、表示しています。次ページに、キャプチャ画像を示します。なかなかそれっぽいでしょう。

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

オブジェクトを ui から追加する 1

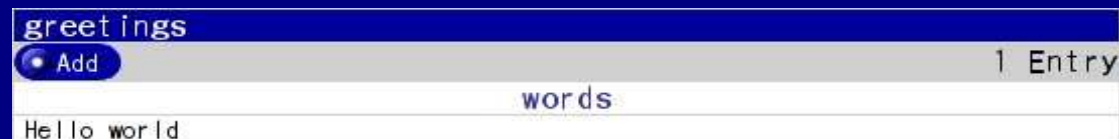
さて、Greeting オブジェクトを ui から追加する為にはまず「追加」ボタンから作りましょう。
右下のような「追加」ボタンが現われます。

```
$ 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();
?>
```



オブジェクトを ui から追加する 2

次は追加するためのフォームを作成します。右下のようなフォームが出来ます。

```
$ 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 text input field labeled "words" and a "Save" button. There is also a "Cancel" button. The form is styled with a blue header and a white body.

オブジェクトを ui から追加する 3

次はフォームからサブミットされた値を処理する php を作成します。赤字の部分でオブジェクトが実際に作られます。

```
$ 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();
?>
```

フォームで保存した文字列が一覧画面にきちんと表示されました。

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	

ハンドラの登録 1

最後に オブジェクトにハンドラを登録します。今回はオブジェクトの生成時と削除時に作動するハンドラを作ります。 glue/conf/greeting.conf でハンドラの定義をします。

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

オブジェクトの生成時に作動するハンドラを記述します。(長いです。次ページに続く)

```
$ vi glue/handlers/handle_greeting.pl
#!/usr/bin/perl -w -I/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);
```

ハンドラの登録 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);
}
```

ハンドラの登録 3

オブジェクトの削除時に作動するハンドラを記述します。

```
$ 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);
}
```

ハンドラの登録 4

この状態で、make rpm 及び cced の reload を行なうと、オブジェクトの生成時、削除時に /etc/greeting.conf が更新されるはずですが、今回は、ui からの削除は未実装なので、コマンドラインツールで削除します。(危険ですけど)

```
$ /usr/sausalito/bin/cceclient
100 CSCP/0.80
200 READY
auth admin *****
109 SESSIONID EnFdfLi3P0NzzDfGdMMzRE5sEy9yat0nW3rx9iaZWympDTrgvOVZl dr6dotdHj
201 OK
find Greeting
104 OBJECT 42
104 OBJECT 43
201 OK
destroy 42
201 OK
destroy 43
201 OK
bye
202 GOODBYE
```

これで、Greeting オブジェクトは一つも無い状態になりました。web の ui では以下のようにになります。



/etc/greeting.conf の中身は空です

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

ハンドラの登録5

もう一度、オブジェクトを追加してみます。

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

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

/etc/greeting.confはどうなったでしょうか？

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

いいですねー。

ハンドラの登録 6

オブジェクトの削除をして、`/etc/greeting.conf` から文字列が削除されれば目標達成ですね。

```
$ /usr/sausalito/bin/cceclient
100 CSCP/0.80
200 READY
auth admin *****
109 SESSIONID uXxurRGxdY4F0tI5jan1Vhk6pyQOKiL248kINQHRFBkhTQc2QoSCv2wIrcmZlWR
201 OK
find Greeting
104 OBJECT 42
201 OK
destroy 42
201 OK
bye
202 GOODBYE

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

駆け足になってしまいましたが、目標達成とします。