给 asterisk 写 app 供 CLI 调用

版本：1.0.0
作者：Mike
E-mail：Mike_Zhang@live.com

环境：CentOS6.2 + Asterisk 1.8.7.1

一、添加源文件

复制 app_verbose.c 为 app_testApp.c
复制 app_verbose.exports 为 app_testApp.exports
主要是修改一些标识，编译不会出错就行，这里列出我进行的主要修改。

1、添加头文件

```
#include "asterisk/cli.h"
```

2、修改变量

```
static char *app_testApp = "testApp";
static char *app_testApplog = "testAppLog";
```

3、在 load_module 中进行注册

```
res |= ast_register_application_xml(app_testApp, testApp_exec);
```

4、添加功能函数

```
static int testApp_exec(struct ast_channel *chan, const char *data)
{
	ast_verb(2,"testApp_exec : %s\r\n",data);
	return 0;
}
```

5、添加 cli 调用接口

注册 command： `e->command = "testApp {print}";`

调用 command：

```
  if (!strcasecmp(a->argv[1], "print")) {
          testApp_exec(chan, a->argv[2]);
      }
```

6、在 unload_module 中进行反注册

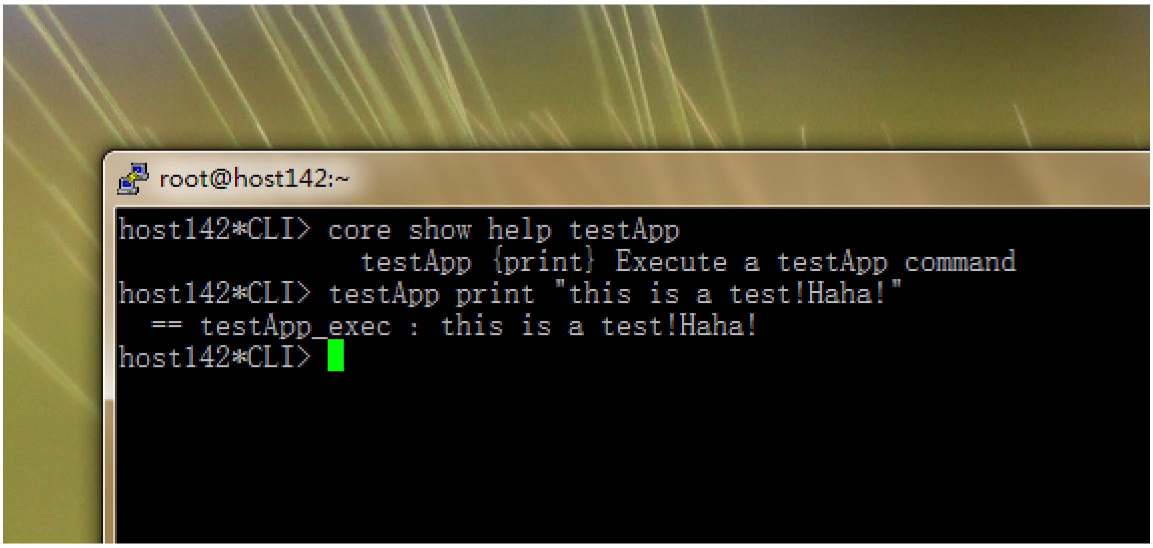
```
res = ast_unregister_application(app_testApp);
```

二、编译并安装

asterisk -rx "core stop now" && make && make install && asterisk && asterisk -rvvvvvvvvvvvvv

三、测试

输入：core show help testApp
会输出帮助文档
输入：core show help testApp "something to test!"



```
host142*CLI> core show help testApp
                 testApp {print} Execute a testApp command
host142*CLI> testApp print "this is a test!Haha!"
  == testApp_exec : this is a test!Haha!
host142*CLI>
```

附源码:

app_testApp.exports 内容为：

```
{
        local:
                *;
};
```

修改如下：

app_testApp.c 内容:

```
 /*Test application : Add by Mike*/

#include "asterisk.h"

ASTERISK_FILE_VERSION(__FILE__, "$Revision: 328209 $")

#include "asterisk/module.h"
#include "asterisk/app.h"
#include "asterisk/channel.h"
#include "asterisk/cli.h"

static char *app_testApp = "testApp";
static char *app_testApplog = "testAppLog";

static int testApp_exec(struct ast_channel *chan, const char *data)
{
	ast_verb(2,"testApp_exec : %s\r\n",data);
	return 0;
}

static int log_exec(struct ast_channel *chan, const char *data)
{
	ast_verb(2,"testApp Module : log_exec!\r\n");
	return 0;
}


static char *handle_cli_testApp(struct ast_cli_entry *e, int cmd, struct
ast_cli_args *a)
{
	struct ast_channel *chan;
```

```
        if(CLI_INIT == cmd) {
                e->command = "testApp {print}";
                e->usage =
                        "Usage: testApp <print> <something2print>\n"
                        "       Print something to test application\n"
                        "       application when the 'print' command is used.\n";
                return NULL;
        }

        if (a->argc < 2)
                return CLI_SHOWUSAGE;

        if (!strcasecmp(a->argv[1], "print")) {
                testApp_exec(chan, a->argv[2]);
        }else{
                return CLI_SHOWUSAGE;
        }

        return CLI_SUCCESS;
}

static struct ast_cli_entry cli_testApp[] = {
        AST_CLI_DEFINE(handle_cli_testApp, "Execute a testApp command")
};

static int unload_module(void)
{
        int res;

        ast_cli_unregister_multiple(cli_testApp, ARRAY_LEN(cli_testApp));

        res = ast_unregister_application(app_testApp);
        res |= ast_unregister_application(app_testApplog);

        return res;
}

static int load_module(void)
{
        int res;

        ast_cli_register_multiple(cli_testApp, ARRAY_LEN(cli_testApp));

        res = ast_register_application_xml(app_testApplog, log_exec);
        res |= ast_register_application_xml(app_testApp, testApp_exec);

        return res;
}

AST_MODULE_INFO_STANDARD(ASTERISK_GPL_KEY, "testApp by Mike");
```