扩展 Asterisk1.8.7 的 Dialplan Applications

我前面有一篇文章介绍了怎么扩展 Asterisk 1.8.7 的 CLI 接口（http://www.cnblogs.com/MikeZhang/archive/2012/06/05/asterisk187CLIAddOns.html），这里说说添加 Dialplan Applications 的方法。具体如下：

一、准备工作

1、进入之前建立的 addons_test 目录，建立 app_testApp20120607.c 和 app_testApp20120607.exports 文件；

2、仿照 app_testApp.c 文件进行 app_testApp20120607.c 的编码，仿照 app_testApp2.exports 文件进行 app_testApp20120607.exports 的编码；

二、编码关键点

1、定义 app 名称为 testApp20120607

2、功能处理函数：

AST_DECLARE_APP_ARGS 函数和 AST_STANDARD_APP_ARGS 函数

需要引入如下头文件：

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

3、核心代码：

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

	AST_DECLARE_APP_ARGS(args,
		AST_APP_ARG(par1);
		AST_APP_ARG(par2);
	);

	char *parse;
	if (ast_strlen_zero(data))
	{
		ast_log(LOG_WARNING, "testApp20120607 requires an argument
(data)\n");
		return -1;
	}
	parse = ast_strdupa(data);
	AST_STANDARD_APP_ARGS(args, parse);

	ast_verb(2,"testApp_exec : par1 = %s\r\n",args.par1);
	ast_verb(2,"testApp_exec : par2 = %s\r\n",args.par2);

	return 0;
}
```

三、测试

1、在拨号方案中调用

vi /etc/asterisk/extensions.conf

在拨号方案中添加如下调用：

exten => _X.,1,testApp20120607(${EXTEN},"Just a test")

2、用呼叫进行触发

用 1003 呼叫 1100，会有如下效果：

```
  == Using SIP RTP CoS mark 5
    -- Executing [1100@DLPN_DialPlan1:1] testApp20120607("SIP/1003-00000008", "1100,"Just a test"") in new stack
  == testApp_exec : 1100,"Just a test"
  == testApp_exec : par1 = 1100
  == testApp_exec : par2 = "Just a test"
E-Mail : Mike_Zhang@live.com
```

说明：

这只是在 CLI 调用程序上增加了接口，我们通过 CLI 接口也可以使用：

```
host232*CLI> testApp20120607 print test,t
  == testApp_exec : test,t
  == testApp_exec : par1 = test
  == testApp_exec : par2 = t
host232*CLI>
E-Mail : Mike_Zhang@live.com
```

完整源码如下：

```
#include "asterisk.h"

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

static char *app_testApp = "testApp20120607";
static char *app_testApplog = "testAppLog20120607";

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

	AST_DECLARE_APP_ARGS(args,
		AST_APP_ARG(par1);
		AST_APP_ARG(par2);
	);

	char *parse;
	if (ast_strlen_zero(data))
	{
		ast_log(LOG_WARNING, "testApp20120607 requires an argument
(data)\n");
		return -1;
	}
	parse = ast_strdupa(data);
	AST_STANDARD_APP_ARGS(args, parse);

	ast_verb(2,"testApp_exec : par1 = %s\r\n",args.par1);
	ast_verb(2,"testApp_exec : par2 = %s\r\n",args.par2);

	return 0;
}

static int log_exec(struct ast_channel *chan, const char *data)
```

```
{
	ast_verb(2,"testApp20120607 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 = "testApp20120607 {print}";
		e->usage =
			"Usage: testApp20120607 <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 testApp20120607 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, "testApp20120607 by
Mike_Zhang@live.com");
```