

apxs - Apache 扩展工具

apxs是一个为Apache HTTP服务器编译和安装扩展模块的工具，用于编译一个或多个源程序或目标代码文件为动态共享对象，使之可以用由mod_so提供的LoadModule指令在运行时加载到Apache服务器中。

因此，要使用这个扩展机制，你的平台必须支持DSO特性，而且Apache httpd必须内建了mod_so模块。apxs工具能自动探测是否具备这样的条件，你也可以自己用这个命令手动探测：

```
$ httpd -l
```

该命令的输出列表中应该有mod_so模块。如果所有这些条件均已具备，则可以很容易地借助apxs安装你自己的DSO模块以扩展Apache服务器的功能：

```
$ apxs -i -a -c mod_foo.c
gcc -fpic -DSHARED_MODULE -I/path/to/apache/include -c mod_foo.c
ld -Bshareable -o mod_foo.so mod_foo.o
cp mod_foo.so /path/to/apache/modules/mod_foo.so
chmod 755 /path/to/apache/modules/mod_foo.so
[activating module 'foo' in /path/to/apache/etc/httpd.conf]
$ apachectl restart
/path/to/apache/sbin/apachectl restart: httpd not running, trying to start
[Tue Mar 31 11:27:55 1998] [debug] mod_so.c(303): loaded module foo_module
/path/to/apache/sbin/apachectl restart: httpd started
$ -
```

其中的参数files可以是任何C源程序文件(.c)、目标代码文件(.o)、甚至是一个库(.a)。apxs工具会根据其后缀自动编译C源程序或者连接目标代码和库。但是，使用预编译的目标代码时，必须保证它们是地址独立代码(PIC)，使之能被动态地加载。如果使用GCC编译，则应该使用-fpic参数；如果使用其他C编译器，则应该查阅其手册，为apxs使用相应的编译参数。

有关Apache对DSO的支持的详细信息，可以阅读mod_so文档，或者直接阅读src/modules/standard/mod_so.c源程序。

语法

```
apxs -g [ -S name=value ] -n modname
```

```
apxs -q [ -S name=value ] query ...
```

```
apxs -c [ -S name=value ] [ -o dsofile ] [ -I incdir ] [ -D name=value ] [ -L libdir ] [ -l libname ] [
-Wc,compiler-flags ] [ -Wl,linker-flags ] files ...
```

```
apxs -i [ -S name=value ] [ -n modname ] [ -a ] [ -A ] dso-file ...
```

```
apxs -e [ -S name=value ] [ -n modname ] [ -a ] [ -A ] dso-file ...
```

选项

一般选项

-n modname

它明确设置了-i(安装)和-g(模板生成)选项的模块名称。对-g选项，它是必须的；对-i选项，apxs工具会根据源代码判断，或(在失败的情况下)按文件名推测出这个模块的名称。

查询选项

-q

查询某种apxs设置的信息。该选项的query参数可以是下列一个或多个字符

串：CC, CFLAGS, CFLAGS_SHLIB, INCLUDEDIR, LD_SHLIB, LDFLAGS_SHLIB, LIBEXECDIR, LIBS_SHLIB, SBINDIR, SYSCONFDIR, TARGET

。这个参数用于手动查询某些设置。比如，要手动处理Apache的C头文件，可以在Makefile中使用：

```
INC=-I`apxs -q INCLUDEDIR`
```

配置选项

-S name=value

此选项可以改变apxs的上述设置。

模板生成选项

-g

此选项生成一个名为name的子目录(见选项-n)和其中的两个文件：一个是名为mod_name.c的样板模块源程序，可以用来建立你自己的模块，或是学习使用apxs机制的良好开端；另一个则是对应的Makefile，用于编译和安装此模块。

DSO编译选项

- c
此选项表示需要执行编译操作。它首先会编译C源程序(.c)files为对应的目标代码文件(.o), 然后连接这些目标代码和files中其余的目标代码文件(.o和.a), 以生成动态共享对象dsofile。如果没有指定-o 选项, 则此输出文件名由files中的第一个文件名推测得到, 也就是默认为mod_name.so。
- o dsofile
明确指定所建立的动态共享对象的文件名, 它不能从files文件列表中推测得到。如果没有明确指定, 则其文件名将为mod_unknown.so。
- D name=value
此选项直接传递到给编译命令, 用于增加自定义的编译变量。
- I incdir
此选项直接传递到给编译命令, 用于增加自定义的包含目录。
- L libdir
此选项直接传递到给连接命令, 用于增加自定义的库文件目录。
- l libname
此选项直接传递到给连接命令, 用于增加自定义的库文件。
- Wc, compiler-flags
此选项用于向编译命令 libtool --mode=compile 中附加compiler-flags, 以增加编译器特有的选项。
- Wl, linker-flags
此选项用于向连接命令 libtool --mode=link 中附加linker-flags, 以增加连接器特有的选项。

DSO的安装和配置选项

- i
此选项表示需要执行安装操作, 以安装一个或多个动态共享对象到服务器的modules目录中。
- a
此选项自动增加一个LoadModule行到httpd.conf文件中, 以激活此模块, 或者, 如果此行已经存在, 则启用之。
- A
与 -a 选项类似, 但是它增加的LoadModule命令有一个井号前缀(#), 即此模块已经准备就绪但尚未启用。
- e
表示需要执行编辑操作, 它可以与 -a 和 -A 选项配合使用, 与 -i 操作类似, 修改Apache的httpd.conf文件, 但是并不安装此模块。



举例

假设有一个扩展Apache功能的模块mod_foo.c, 使用下列命令, 可以将C源程序编译为共享模块, 以在运行时加载到Apache服务器中:

```
$ apxs -c mod_foo.c
/path/to/libtool --mode=compile gcc ... -c mod_foo.c
/path/to/libtool --mode=link gcc ... -o mod_foo.la mod_foo.slo
$ _
```

然后, 必须修改Apache的配置, 以确保有一个LoadModule指令来加载此共享对象。为了简化这一步骤, apxs可以自动进行该操作, 以安装此共享对象到"modules"目录, 并更新httpd.conf文件, 命令如下:

```
$ apxs -i -a mod_foo.la
/path/to/instdso.sh mod_foo.la /path/to/apache/modules
/path/to/libtool --mode=install cp mod_foo.la /path/to/apache/modules ... chmod 755
/path/to/apache/modules/mod_foo.so
[activating module 'foo' in /path/to/apache/conf/httpd.conf]
$ _
```

如果配置文件中尚不存在, 会增加下列的行:

```
LoadModule foo_module modules/mod_foo.so
```

如果你希望默认禁用此模块, 可以使用 -A 选项, 即:

```
$ apxs -i -A mod_foo.c
```

要快速测试apxs机制, 可以建立一个Apache模块样板及其对应的Makefile:

```
$ apxs -g -n foo
Creating [DIR] foo
Creating [FILE] foo/Makefile
Creating [FILE] foo/modules.mk
Creating [FILE] foo/mod_foo.c
Creating [FILE] foo/.deps
$ _
```

然后，立即可以编译此样板模块为共享对象并加载到Apache服务器中：

```
$ cd foo
$ make all reload
apxs -c mod_foo.c
/path/to/libtool --mode=compile gcc ... -c mod_foo.c
/path/to/libtool --mode=link gcc ... -o mod_foo.la mod_foo.slo
apxs -i -a -n "foo" mod_foo.la
/path/to/instldso.sh mod_foo.la /path/to/apache/modules
/path/to/libtool --mode=install cp mod_foo.la /path/to/apache/modules ... chmod 755
/path/to/apache/modules/mod_foo.so
[activating module 'foo' in /path/to/apache/conf/httpd.conf]
apachectl restart
/path/to/apache/sbin/apachectl restart: httpd not running, trying to start
[Tue Mar 31 11:27:55 1998] [debug] mod_so.c(303): loaded module foo_module
/path/to/apache/sbin/apachectl restart: httpd started
$ _
```