服务器知识手册（五）

狄登峰 主编

需要全本请访问 www.ertongbook.com

目　　录

Web 服务器.....1
浅析 Apache 中 SSI 和 CGI 的设定方法 1
Apache 与 Tomcat 安装配置手册 6
Resin 在 Windows 系统下的安装 10
给 IISWeb 服务器装上一把锁 13
安全配置和维护 ApacheWEBServer 27
Apache 服务器的保护 34
优化 Apache 服务器的重负荷 39
用反向代理技术保护 Web 服务器 40
用 Apache 架设 Web 服务器 45
实现四台 Web 服务器的负载均衡 57
使用负载均衡技术建设高负载的网络站点 62
Web 服务器负载均衡方案 68
关于双连接的负载均衡 73
实现 Web 中的@虚拟域名系统（原理篇） 80
利用 Apache+PHP3+MySQL 建立动态网站 85
10 分钟分析 Web 服务器 93
多服务器的日志合并统计 95
构建 SCOUNIX 下的 Web 服务器 106
AppML--Web 开发的未来 110
在 WebService 中使用 ASP.net 状态保持 113

Web 服务器

浅析 Apache 中 SSI 和 CGI 的设定方法

由于 Apache 具有相当高的可移植性，它支持超过 30 种操作系统，包括 Unix、Windows 及 Darwin 等系统，所以目前在网络上已注册的网域里大部份是使用 Apache 网页服务器。目前 ApacheSoftwareFoundation 正致力于发展现在已进入 alpha 测试阶段的 Apache2.0。在这里，我和大家探讨如何修改服务器选项让服务器能提供简单的动态网页内容，也就是支持 CGI 程序及 Server-SideInclude（SSI）程序。

1、准备工作

首先，我假设你已经安装好 Apache 而且你的 Apache 能提供静态网页供浏览。Apache 的安装会自动附上静态的 HTML 测试页，也就是说如果能看到那测试页，就代表你的 Apache 能正常运作了。基本安装下的 Apache 仅能提供静态的 HTML 网页。然而，你可以通过使用模块（modules）来提升它的功能。在原始的设定下，Apache 的编译会包含 mod_include 及 moc_cgi 这两个模块。你可以在 bin 子目录下执行./httpd-l 来查看你的 Apache 是否装有这两个模块。执行的输出会是一长串 Apache 现在安装的所有模块。如果 mod_include 及 moc_cgi 这两个模块不在清单里，你必须重新编译服务器。重新编译时，确定依照如何含括 mod_include、moc_cgi 模块的说

明。此外你还必需有提供实时网页的 Server-SideIncludes（相关资料网址：http://www.oreilly.com/catalog/apache/excerpt/ch10.html）。有了 server-sideincludes（SSI）支持，你就能制作出实时的动态网页。接下来，我先从服务器的 SSI 支持设定开始，然后进入 CGI 的编写。

2、Apache 的设定

首先你必须先找到 Apache 这个设定文件。Apache 的原始安装目录在 Unix 下是/usr/local/apache，在 Windows 下则是 c:\ProgramFiles\Apache。接着在 conf 子目录下你会找到 httpd.conf 文件。这就是 Apache 的设定档。这个设定文件是个纯文字文件，所以你可以使用一般的文字编辑器，如 vi 或 Notepad 等，来编辑。首先要注意的是在这个设定档里有些行的起始文字是#符号，这表示这行的文字全为批注。适当地在你的设定档内做批注是个好习惯，因为那帮你记得你曾做了哪些设定以及为什么。

3、执行 SSI 程序

开启设定文件并寻找以下这些文字：

```
#
#Touseserver-parsedHTMLfiles
#
#AddTypetext/html.shtml
#AddHandlerserver-parsed.shtml
```

删除 AddType 及 AddHandler 这两行指令前的#符号。AddType 指令会要求服务器在传回任何附属档名为.shtml 的网页时，以 text 或 HTML 做为传回文件的内容格式。AddHandler 则是用来指示服务器将文件内容送

交给 mod_include 处理。之后，mod_include 就会判断该如何响应这样的文件。接下来，寻找以下文字：

在这行文字及对应的间会有一行选项行（optionsline）。原始的设定是：

OptionsIndexesFollowSymLinksMultiViews

在这行尾端加上 Includes，结果看起来会是这样：

OptionsIndexesFollowSymLinksMultiViewsIncludes

这是要求 Apache 在 htdocs 子目录里执行 server-sideincludes 程序。为了让这些修改生效，我们必须重新启动服务器。在 Unix 下重新激活，执行 "kill-HUP‘cat/usr/local/apache/logs/httpd.pid‘" 。在 Windows 下，执行"Apache-krestart"。现在我们来试试刚才的设定结果。在/usr/local/apache/htdocs 目录里新增一个文件 test.shtml。这个文件必须要包含以下程序代码：

Thefilehello.txtislonganditwaslastmodifiedon

这段 SSI 程序会去读取一个称为 hello.txt 文件，并将该文件的大小以及最近一次的修改日期输出到网页上。显然的，我们还必须在 htdocs 目录下新增这个 hello.txt 文件。在我的 hello.txt 文件里只有一行文字：HOWAREYOU!。完成新增这些文件后，打开你惯用的浏览器并开启 http://localhost/test.shtml 网页。如果你服务器的安装并不是通过 root 用户，你可能必须改为开启 http://localhost:8080/test.shtml。之后将得到如下结果：

HOWAREYOU!Thefilehello.txtis1kbyteslonganditwaslastmodifiedonWednesday,02-Aug-200020:18:28PDT

另外一种可以激活支持 SSI 程序的方法称为 XbitHack 设定（相关资料网址：http://www.apache.org/docs/mod/mod_include.html#xbitha

需要全本请访问 www.ertongbook.com

ck)。这个方法的由来是当你将文本文件的使用者可执行位（user-executablebit）设为可执行状态后，Apache 会将那些文件视为 SSI 程序文件。

要激活这样的功能必须将以下指令（directive）放在所有目录的 .htaccess 文件里：XbitHackstatuson(orfull)status 的值可以设为 on、off 或是 full。on 的设定会强制服务器将所有使用者可执行的文件视为 SSI 项。Off 则使服务器完全忽略使用者可执行的设定状态。若是设定为 Full，服务器会视所有使用者可执行档为 SSI 项，同时也会检查组可执行（group-executablebit）。如果组可执行项设定为可执行时，传回 header 的 lastmodifiedtime 的值就会被设定为该文件最近一次被修改的时间。这样的设定可以让客户端的浏览器及代理服务器（proxy）进行缓存（caching）。不过在使用这样的功能时必须要小心。例如说，如果你的网页有提供轮替式广告看板你就不会想要设定群组可执行位为开启的状态，因为那么做会让第一个下载的广告被快取起来，导致使用者再也看不到其它页的广告。

4、执行 CGI 程序

在 Apache 原始安装里，cgi-bin 子目录下附有两组 CGI 程序，test-cgi 以及 printenv，只不过这两组程序有潜在的安全漏洞。但是由于我们只是要做设定测试，并且我们不会将这样的原始安装设定直接放在主运行服务器（liveserver），所以我们还是会激活其中一组 CGI 程序，看看 Apache 当初是如何被设定来执行这组程序。最后我们会自己撰写一支简单的 CGI 程序。

首先，要确定这组程序是能执行的。进入 cgi-bin 子目录，确定程序文件被设定为使用者（服务器执行时使

用者）可执行以及使用组（服务器执行时使用组）可执行。对 Windows 系统来说，这一步应该是非必要的。接着，对服务器要求这样的内容：

http://localhost:8080/cgi-bin/test-cgi

注意：只有在通过非 root 使用者进行服务器安装的情况下才需要指定 8080 端口（port）。这支 Apache 内建的 test-cgi 程序会列出 CGI 程序会存取的变量值。激活 CGI 支持是设定在 httpd.conf 设定文件内的 ScriptAlias 指令区段。这个指令区段的原始设定值是：

ScriptAlias/cgi-bin/"/usr/local/apache/cgi-bin/"

这行指令是告诉 Apache 如果要求的网页路径是以 cgi-bin 为起始，这些文件可在/usr/local/apache/cgi-bin/目录下找得到。这行指令同时也告诉 Apache 要在这个目录下执行文件。在下面我准备了一个会输出"Howareyou!"的简单 CGI 程序。我将它命名为 how.sh。

```
#!/bin/sh
echo"Content-type:text/html"
echo
echo"Howareyou!"
```

虽然这组 CGI 是采用 shellscript 来编写，其实它可以用任何适用于该系统的语言来撰写。至于关于 CGI 程序的撰写如果有机会我会和大家作更深入的探讨。

结论：

Apache 支持的所有 SSI 指令可以在 Apachedocumentation 里找到，Apache 所有的功能都可以通过设定文件（configfile）进行调试。在这里我所介绍的仅只是设定文件相关知识的皮毛。设定文件的原始设定有着非常详尽的说明文件，而且每一个系统版本都

附有核心模块及标准模块的说明文件，如果你花些时间在这些文件里摸索，你会找到任何你想要的功能。

Apache 与 Tomcat 安装配置手册

1.所需的软件包

j2sdk-1_4_1_02-linux-i586

Apache2.0.40(RedHatLinux)

jakarta-tomcat-4.1.18

jakarta-tomcat-connectors-4.1.18

2.软件包下载地址

(1)JavaJ2SE1.4.1

URL：http：//java.sun.com/j2se/1.4.1/download.html

(2)Apache2.0.40

URL：http：//httpd.apache.org

(3)Tomcat4.1.18

URL：http：//jakarta.apache.org/builds/jakarta-tomcat-4.0/release/v4.1.18/bin/jakarta-tomcat-4.1.18.tar.gz

(4)Tomcatconnectors-4.1.18

URL：http：//jakarta.apache.org/builds/jakarta-tomcat-4.0/release/v4.1.18/src/jakarta-tomcat-connectors-4.1.18-src.tar.gz

3.软件安装

(1)解开压缩包

使用 gunzip 或 tar 或 rpm 解开下载的软件包，按照默认方式安装既可。

(2)目录的处理

J2SDK 安装目录：/usr/j2sdk1.4.1_02

Apache **安装目录**：/etc/httpd

/usr/lib/apache

/var/www

Jakarta-Tomcat **安装目录**：/opt/jakarta-tomcat-4.1.18

Jakarta-Tomcat-connectors **安装目录**：/opt/jakarta-tomcat-connectors-4.1.18-src

4.**环境变量设置**

JAVA_HOME=/usr/j2sdk1.4.1_02

CLASSPATH=$JAVA_HOME/lib：$JAVA_HOME/jre/lib

CATALINA_BASE=/opt/jakarta-tomcat-4.1.18

CATALINA_HOME=/opt/jakarta-tomcat-4.1.18

PATH=$JAVA_HOME/bin：$JAVA_HOME/jre/bin：$PATH：$HOME/bin

5.**特殊安装**

(1)**编译** Jakarta-Tomcat-connectors

cd/opt/jakarta-tomcat-connectors-4.1.18-src/webapp##**进入指定的目录**

./support/buildconf.sh##**生成编译配置文件**

./configure--with-apxs##**生成** Makefile **编译文件**

make##**编译生成** mod_webapp.so

注释：生成的 mod_webapp.so **文件在** apache-2.0 **目录中**

提醒：如果生成 mod_webapp.so **时失败，并提示找不到** apxs，**则证明你的** Apache **在安装的时候，未安装** Apache **附加的开发包，请下载安装该开发包，则会解决该问题(这个包在** RedHat8 **中的文件名是**：httpd-devel-2.0.40-8.i386.rpm)

(2)把 apache-2.0 目录中的 mod_webapp.so 文件复制到目录/usr/lib/httpd/modules 并检查文件/usr/lib/httpd/modules/mod_webapp.so 的属性，应与其它 Modules 文件属性一致，如果不相同，则使用 chmodchgrpchown 来修改。

6.独立环境的测试

(1)测试 Java 的运行状况，如下命令：

java-version

javac-version

看到版本号则表示 JDK 安装成功

(2)测试 Apache 的运行状况，如下命令：

/etc/init.d/httpdrestart##重新启动 ApacheServer

使用任意一浏览器，输入服务器地址后回车，应该可以看到 Apache 的默认主页

(3)测试 Jakarta-Tomcat 的运行状况，如下命令：

cd$CATALINA_HOME

cdbin

./startup.sh##启动 jakarta-tomcat-4.1.18 服务

使用任意一浏览器，输入服务器地址和端口号 8080 后回车，应该可以看到 jakarta-tomcat-4.1.18 的默认主页

例如：http：//192.192.192.1：8080

./shutdown.sh##关闭 jakarta-tomcat-4.0.1 服务

7.整合安装设置

(1)修改/etc/httpd/conf/httpd.conf 文件

在#DynamicSharedObject(DSO)Support 之后，添加如下内容

LoadModulewebapp_modulemodules/mod_webapp.so

(2)测试配置和 Modules，执行以下命令

apachectlconfigtest

如果出现 SyntaxOK，则证明 Module 安装和配置成功了!

3)修改/etc/httpd/conf/httpd.conf 文件

在文件尾部添加如下内容

WebAppConnectionwarpConnectionwarplocalhost：8008

WebAppDeployexampleswarpConnection/examples/

4)再次测试配置，执行以下命令

apachectlconfigtest

如果出现 SyntaxOK，则证明安装成功!

如果出现以下错误信息

Syntaxerroronline1038of/etc/httpd/conf/httpd.conf：

Invalidvirtualhostname

则修改 httpd.conf 文件，找到#ServerName 所在的行，并在其下面加入如下内容即可

ServerNamelocalhost：8008#制定服务器的名称和端口

5)检查$CATALINA_HOME/conf/server.xml 文件中相关的配置，默认情况下的

相关的信息是被注释掉了，所以需要删掉注释的语句，这样 Tomcat 在启动的时候才会初始化

相关的内容并开启 8008 端口!

8.整合测试

重新启动所有服务

$CATALINA_HOME/bin/shutdown.sh

$CATALINA_HOME/bin/startup.sh

/etc/init.d/httpdrestart

在浏览器中输入如下地址

需要全本请访问 www.ertongbook.com

http：//服务器 IP/examples/jsp/index.html，如果能够显示正常的网页，则安装成功

在浏览器中输入如下地址

http：//服务器 IP/examples/jsp/jsptoserv/jsptoservlet.jsp 如果能够显示正常的网页和时间，则 jsp 和 Servlet 安装成功。

Resin 在 Windows 系统下的安装

JSP 是 Sun 在 servlet 基础上发展而来的一种新的 Web 开发工具，在国外 Ejb+jsp/servlet+应用服务器+数据库已经已经成为电子商务站点的流行架构。如果您选用 JSP 平台作为 Internet 商业站点的支持，那么速度、价格和稳定性都是要考虑到的，作为这些 Resin 是十分出色的。Resin 在 Linux 下的安装已经有不少文章介绍过，本文介绍它在 Windows 操作系统下的安装，相信看过本文，您将会很轻松架构起一个拥有强大功能的 Web 站点。

首先的要求，是 JDK 已经正常安装。在这里我就不冗述 JDK 的安装了，这方面的文章比较多。

Resin 达到 JSP 解释器的目的，有两种方法，一种是和 IIS 集成，一种是作为 StandAlone 的 Httpd。具体使用那种方法，要看个人喜好和实际情况。我用的是第一种。

首先先讲解最简单得 StandAlone 法。

这种方法几乎不需要配置，将 resin 解压缩之后，直接运行 bin\httpd.exe 即可。需要注意的是，运行 httpd.exe 之后会产生两个窗口，一个是类似命令行窗口（但是

只能输出，不能输入），另外一个是 GUI 的控制窗口，这两个窗口绝对不可以用直接 X 的方法关闭，这样会导致 Resin 的系统错误，正确的方法是用 Gui 上面的 quit 按钮。

等到命令行窗口里面出现：

Resin2.1.6--MonOct217:34:31PDT2000

listeningto*:8080

这样就是启动成功了，后面的 8080 就是 resinhttpd 现在所监听的端口了。这个端口是可以修改的。修改的时候，需要改动 conf\resin.conf 文件，修改如下

Section:XML:

<!--thehttpport-->

<httpd-port>8080</httpd-port>

把这里的 8080 修改成你想要得 port 即可。但是请注意不要和已经占用的端口重复，否则无法启动。这个时候，resinhttpd 已经在接受访问了，具有 IIS 的基本功能，httproot 目录默认是 resin 安装目录的 doc 目录但是也是可以修改的，可以修改

conf\resin.conf:XML:

<http-server>

<app-dir>doc</app-dir>

这里是默认的情形，你也可以写上绝对路径来定义其他的目录。输入 http://localhost:8080/看看！

再说一下如何同 IIS 集成。在这种模式下，resin 作为 IIS 的插件运行，只有碰到.jsp，.xtp 等等文件的时候才进行解析，就好像 asp.dll 负责解析.asp 文件，perl.dll 负责解析.pl 文件一样。这个配置比较复杂，而且可能会导致问题，但带来非常大的方便。

请三思而后行。

下面我要说的是手工安装的办法，个人感觉它提供的 setup.exe 安装起来常常有问题，而且不可靠。

1.安装好 jdk,resin 之后（这些都是废话）

2.执行 bin\setup.exe 文件

填好关于 resinhome 以及 iis/pws 目录之后，点击 OK，他会停止 IIS/复制 iis_run.dll/启动 IIS 这个时候，在你的 IIS 的 inetpub\scripts 目录下面会有一个 iis_run.dll 文件。同时在 IIS 配置的 filter 里面会多出来一个 SRun。

3.这个时候 resin 还不能使用，还需要安装一个服务执行 bin\目录下面的 srun.exe-install 命令。这个会给系统安装一个 Resin 服务。

4.配置 resin.conf 文件

编辑如下 Section:

<http-server>

<app-dir>doc</app-dir>

将 doc 修改为你的 wwwroot 所在的目录，这个时候就好了！

5.重新启动系统

这一步是必需的，若不重新启动，Resin 服务会无法正确运行。

6.浏览 http://localhost/caucho-status 看看！

这样的话，大致就告一段落了。以后，IIS 和 resin 会各司其职，IIS 负责 html 和 asp 文件的解析，resin 负责 jsp 文件的解析。最后要说的是，resin 的功能非常强大，虚拟目录、虚拟主机、负载均衡都是可以实现的。

给 IISWeb 服务器装上一把锁

为了提高 IIS 的安全性，微软提供了两个工具：IIS Lockdown 和 URLScan，其中 IISLockdown2.1 包含了 URLScan。IISLockdown2.1 具有如下功能：

•禁用或者删除不必要的 IIS 服务和组件。

• 修改默认配置，提高系统文件和 Web 内容目录的安全性。

•用 URLScan 来过滤 HTTP 请求。

本文介绍如何运用 IISLockdown2.1 的前两项功能。注意本文的说明针对 IISLockdown2.1 版本，以前版本的用法大不相同。

一、注意事项

IISLockdown 会改变 IIS 的运行方式，因此很可能与依赖 IIS 某些功能的应用冲突。特别地，如果要在一个运行 MicrosoftExchange2000Server、ExchangeServer5.5 或 MicrosoftSharePointPortalServer 的服务器上安装 IISLockdown 和 URLScan，应当加倍小心。

微软的两篇文章解释了可能遇到的困难和解决办法：《XADM：在 Exchange2000 环境中使用 IISLockdown 向导的已知问题和调整策略》（http://support.microsoft.com/default.aspx?scid=kb;en-us;q309677），以及《SPS：IISLockdown 工具影响 SharePointPortalServer》（http://support.microsoft.com/default.aspx?scid=kb;en-us;q309675）。

另外，在正式应用 IISLockdown 或 URLScan 之前，务必搜索微软的知识库，收集可能出现问题的最新资料。

掌握这些资料并了解其建议之后，再在测试服务器上安装 IISLockdown，全面测试 Web 应用需要的 IIS 功能是否受到影响。最后，做一次全面的系统备份，以便在系统功能受到严重影响时迅速恢复。

二、安装

IISLockdown2.1 可以从 http://www.microsoft.com/downloads/release.asp?releaseid=33961 下载。下载之后得到一个 iislockd.exe，双击运行，把它解压缩到一个临时目录并启动 IISLockdown 向导。但是，如果要用 IISLockdown 来保护多个服务器，最好按照下文的说明把它解压缩到一个专用目录，这样就不必每次运行 IISLockdown 都要重新解压缩了。

必须注意的是，下载得到的是一个自解压缩的执行文件，这个执行文件与压缩包里面的应用执行文件同名。因此，如果把 iislockd.exe 解压缩到它本身所在的目录，就会引起文件名称冲突。请按照下面的安装步骤执行，以避免可能出现的问题：

㈠将 iislockd.exe 下载到一个临时目录。

㈡打开控制台窗口，进入临时目录，执行命令“iislockd.exe/q/c/t:c:\IISLockdown”解开压缩，/q 要求以“安静”模式操作，/c 要求 IISLockdown 只执行提取文件的操作，和-t 选项一起使用，-t 选项指定了要把文件解压缩到哪一个目录（例如在本例中，要求把文件解压缩到 c:\IISLockdown 目录）。表一列出了 iislockd.exe 解压缩得到的主要文件，注意 iislockd.exe 包含了 URLScan 的文件，但本文不准备详细探讨 URLScan。

表一：IISLockdown2.1 主要文件

IISLockdown 文件	说明
iislockd.exe	IISLockdown 主执行文件。
iislockd.ini	配置和选项文件。
iislockd.chm	联机帮助。
runlockdunattended.doc	有关“无人值守”运行方式的文档。
404.dll	“文件没有找到”应答文件。
URLScan 文件	说明
urlscan.exe	URLScan 安装程序包。
urlscan.doc	URLScan 文档。
urlscan*.ini	配置和选项文件。
urlscan_unattend.txt	无人值守方式安装 URLScan 的配置文件。
readme.txt	针对无人值守方式运行 URLScan 的说明
unattend.cmd	无人值守方式安装 URLScan 的命令文件。

需要全本请访问 www.ertongbook.com