

100003	2	udp	2049	nfs
100003	2	tcp	2049	nfs

这里表明 portmapper、mountd 和 nfsd 已宣称了它们的服务并工作正常。在 Web 服务器上设置 NFS 的客户端之前，下面提及的安全问题是很重要的。

#### 19.6.2.2 服务器安全问题

与 nfsd 组合在一起的 portmapper 可能被欺骗，使它能够没有任何权力的情况下获得 NFS 服务器上的文件。所幸的是，Linux 使用的 portmapper 对攻击是相对安全的。并且可以在 /etc/hosts.deny 文件中增加下列行，使其更为安全：

```
portmap: ALL
```

系统将拒绝任何人访问 portmapper。现在，需要对 /etc/hosts.allow 文件做如下修改：

```
portmap: 192.168.1.0/255.255.255.0
```

这将允许所有来自 192.168.1.0 网络的主机有权访问 portmapper 管理的程序，如 nfsd 和 mountd。



不要在 /etc/hosts.allow 文件中的 portmap 行中使用主机名。因为使用主机名查找将会间接地使 portmap 活动，这将触发循环的主机名查找。

服务器端的另一个安全问题是允许客户机的超级帐号 root 在服务器端也被当作超级帐号 root 对待。在缺省的情况下，Linux 禁止 NFS 客户端的 root 帐号在服务器端也作为 root 对待。换句话说，在服务器上由 root 拥有的输出文件不能被客户端的 root 用户修改。为了明确地强制实施这一规则，可以将 /etc/exports 文件修改如下：

```
/www/cgi-data www1.xcnews-lan.com (rw, root_squash)
www2.xcnews-lan.com (rw, root_squash)
```

现在，如果一个用户使用客户端的 UID 0 (root 用户) 访问 (读，写，或删除) 任何 NFS 文件系统，服务器将用服务的 “nobody” 帐号的 UID 替换它。这意味着客户端的 root 用户不能访问或改变那些只能由服务器上 root 访问或改变的文件。



为了给予客户端的 root 访问 NFS 系统的权限，使用 no\_root\_squash 选项，替代 root\_squash。

到此为止，已经设置了 NFS 服务器，并且是安全的。因此，现在让我们设置 NFS 客户主机。

#### 19.6.2.3 设置 NFS 客户

在缺省情况下，RedHat 支持 NFS 文件系统，因此不需要改变其内核。为了装配由 ns.xcnews-lan.com 输出的 /www/cgi-data 目录，在两个 Web 服务器的 /etc/fstab 文件中添加下列行：

```
ns.xcnews-lan.com:/www/cgi-data /www/cgi-data nfs
```

当任一 web 服务器重新引导时，上面的命令行将自动装配 /www/cgi-data 目录。下面，在两个系统上都生成 /www/cgi-bin 目录，并手工地用 mount 命令装配此目录如下：

```
ns.xcnews-lan.com:/www/cgi-data /www/cgi-data nfs
```

Tip



NFS 装配出现的典型的问题是大多数人在修改 NFS 服务器上的 `/etc/exports` 文件后忘了运行 `exports` (也就是重新启动 `rpc.mountd` 和 `rpc.nfsd`)。

卸下 NFS 文件系统同卸下本地文件系统完全一样。注意,也可以通过不要太信任 NFS 服务器来增强 NFS 客户安全性。例如,你能用 `nosuid` 选项禁止在 NFS 文件系统上使用 `suid` 程序。这表示服务器的 `root` 用户不能在文件系统中使用 `suid-root` 程序,因此不能先作为一个普通用户注册到客户机上,然后又使用 `suid-root` 程序成为客户机上的 `root`。`nosuid` 选项和 `noexec` 选项一起,也可以禁止执行被装配的文件系统上的文件。可以在 `/etc/fstab` 文件中说明 NFS 装配点的行的选项列中加入这些选项。

到此为止,文件分发方案和基于 NFS 的 CGI 数据目录都已准备好了,该是配置 Apache 并确保它安全的时候了。

## 19.7 Apache 服务器配置

Apache 服务器配置和通常一样, XC News 并不需要满足特殊的要求。

由 `makesite` 脚本产生的 `<VirtualHost>` 容器段需要被加到每个 Web 服务器的 `httpd.conf` 文件中。正如前面提到的,这些虚拟主机指定专用于每个 Web 服务器的 IP 地址。使用 `/etc/rc.d/rc3.d/S85httpd` 脚本(也就是 `/etc/rc.d/init.d/httpd` 脚本)检查,以确保在机器引导过程的最后启动 Apache 服务器。

对于需要保证安全的 Web 服务,我竭力推荐在 `secured.xcnews.com` 上使用 Stronghold,这只需要在 `ns.xcnews.com` 中另增加一个 CNAME 项。正如前面提到的,XC News 并不想建立这样一种配置:即每个虚拟主机(客户站点)都要为某个 CA 提供的证书付费。因此在 `secured.xcnews.com` 上建立 Stronghold 是很合理的。在这种配置里,Stronghold 服务器指向 `/www/secured` 作为文档根目录。在这种情况下,XC News 能够在 `/www/secured` 下创建子目录,使客户把那些需要通过基于 SSL 安全通信通道访问的文件存储到这些目录下。例如,如果客户站点 `www.client-a.com` 需要安装称为 `order.html` 安全的订购表单,他们需要做的第一件事就是要求 XC News 在 `/www/secured` 目录下为它们生成一个子目录,并把这个文件存放在那儿。XC News 可以生成一个 `/www/secured/client-a` 子目录,并把 `order.html` 存放在此目录下。现在客户能使所有指向 `order.html` 的引用(A HREF 链接)指向 URL `https://secured.xcnews.com/client-a/order.html`

由于 XC News 希望在 `www.xcnews.com (/www/xcnews/cgi-bin)` 上的 CGI 脚本也通过安全服务器进行访问,所以需要在 Stronghold 的 `httpd.conf` 文件中增加 `ScriptAlias` 命令,如下:

```
ScriptAlias cgi-bin /www/xcnews/cgi-bin
```

这允许安全服务器运行同样的脚本。

客户的文档应当包括返回到原始客户站点的链接。例如,如果一个 CGI 脚本处理了

XC News 的订购表单, 则输出页应包括指向原始客户页面的链接。这可使这种配置变得对用户更为友好。

无论何时, XC News 客户生成的 HTML 页面被放到安全的服务器的子目录里时, 都应检查一下, 看是否有指向原始客户站点的返回链接。如果页面没有这样的链接, 访问者也许就会分不清。

对所有处理 HTML 表单的 CGI 脚本或其他可能的 CGI 脚本, XC News 都应当确保这些脚本有一种方式使访问者可以返回的原始的 Web 站点。例如, 一个 HTML 表单可以包含一个隐含变量, 如下:

```
<INPUT TYPE=HIDDEN NAME="return-url" VALUE="http://www.  
client-a.com/thanks_for_the_order.html")
```

处理此表单的 CGI 脚本应当使用此变量, 或者把变量 `return-url` 的值作为重新定向到用户的 URL, 或者用作生成输出页上明显标明“返回我们的站点”或其他类似的链接。

## 19.8 客户的 FTP 用户帐号

XC News 需要客户通过 FTP 更新他们的 Web 站点。因此, 不必在 `ns.xcnews.com` 服务器上创建 FTP 帐号。由于已经在 DNS 中为 `xcnews.com` 域建立了 `ftp.xcnews.com` 别名, 它能用作正式的 FTP 服务器的 IP 的别名。

RedHat 提供了叫作 `/usr/sbin/adduser` (或 `/usr/sbin/useradd`) 的程序, 可以从命令行用它创建新用户。这个程序使用存储在 `/etc/default/useradd` 文件里的缺省配置。此文件包含的 `home` 目录的缺省设置如下:

```
HOME=/home
```

把它变成 `HOME=/www`, 这样, 只要帐号名和客户域名一样, 每个客户用户帐号都将用它自己的 Web 目录作为 `home` 目录。

Note

`/www` 目录出现在每个用户的 `home` 目录的路径中, 因此需要将它设置为对每个用户都是可读的和可执行的。

例如:

```
adduser client-a
```

这将创建一个称为 `client-a` 的用户, 他的 `home` 目录是 `/www/client-a`。回忆以 `makesite client-a.com` 运行 `makesite` 时创建了 `/www/client-a` 目录。这样, 用户的 `home` 目录就与 Web 目录一致。当这样一个用户使用 FTP 进入 `ftp.xcnews.com` 时, 将自动在他的 `home` 目录 (也就是它的 Web 站点的顶级目录) 里开始会话过程。

在缺省情况下, `adduser` 程序为每个用户创建一个新组, 并且把组加到 `/etc/group` 文件里。因此, 改变每个 `home` 目录的权限是必要的。例如, 在运行命令 `adduser client-a` 后, 必须运行下面的权限设置命令:

```
chown -R client-a.httpd /www/client-a/
```

```
chmod -R 750 /www/client-a/
```

第一条命令确保 client-a 和它的用户组（也称为 client-a）成为 /www/client-a 目录和所有子目录的所有者。第二条命令设置 /www/client-a 站点的所有目录和文件的权限。所有者（client-a）有读、写、执行权限，httpd 组有读和执行权限，其他用户没有权限。httpd 组应当包含 httpd 用户，因为 httpd 用户被 rdist 和其他技术支持软件用作帐号。由于除所有者和 httpd 用户组外，没有人被允许访问用户的 home 目录，因此一个客户不能通过 FTP 看到其他客户的内容。和通常一样，用户口令可通过使用 /usr/bin/passwd 实用程序设置。创建帐号后，应当通过 FTP 测试其上载和下载权限。

## 19.9 测试新系统

到此为止，新系统上的所有服务都已被设置和运行。因此，可以开始测试阶段。为了从远程主机测试新系统，把远程主机上的域名服务器改变成 ns.xcnews.com 的 IP 地址。这使得远程测试主机向 ns.xcnews.com 查寻，它对新站点有相应的 DNS 设置。这使得测试者可以继续测试，而不影响活动站点上的任何事情。

第一组测试使用简单的 TCP/IP 工具，如 ping、nslookup 和 traceroute。例如，从命令行使用 nslookup，测试者能够发现 ns.xcnews.com DNS 服务器是否能给出关于 xcnews.com 或它的任一客户站点的相应信息。

因为有许多域要进行测试，所以，为了减少完成简单测试而进行的人机对话，至少使用原始的方法是必要的。让我们生成一个包含域名列表的文本文件如下：

```
xcnews.com
client-a.com
lient-b.com
...
...
client-z.org
```

现在可以很从容地用各种 UNIX 工具生成批处理命令文本文件。例如，为了测试每个域的 DNS 信息，使用如下命令：

```
cat ./domain-list.txt | awk '{printf("nslookup -query=ns -
timeout=3 %s\n", $1;)}' > ./ns_test
chmod 750 ./ns_test
./ns_test
```

第一行生成一个 ns\_test 文件，它包含类似如下的行：

```
nslookup -query=ns -timeout=3 xcnews.com
nslookup -query=ns -timeout=3 client-a.com
```

第二行把 ns\_test 脚本变为一个可执行的脚本，第三行运行它。它测试了每个域的域名服务器信息。现在，修改第一行中 -query=ns 为 -query=mx，可以很容易地把第一行变为测试每个域的电子邮件交换记录，如下所示：

```
cat ./domain-list.txt | awk '{printf("nslookup -query=mx -
timeout=3 %s\n", $1;)}' > ./mx_test
```

现在, 可以像 `ns_test` 那样将 `mx_text` 变成可执行脚本, 并运行它测试 MX 记录。相似的, 也可以运行脚本通过 HTTP 测试 Web 站点。下面是这样的脚本的例子:

```
cat ./domain-list.txt | awk ' {printf("lynx -dump -head
www.%s\n", $1);}' ./www_test
chmod 750 ./www_test
./www-test > /tmp/www_problems
```

这里, 用了基于文本的 Web 浏览器 `lynx` 对域名列表中的每个 Web 站点发出 HTTP HEAD 请求。如上述的例子一样, 当得到的 `www_test` 脚本运行时, 在 `/tmp` 下生成一个 `www_problems` 文件。如果当 `lynx` 访问 Web 站点主机时, 因为 Web 站点配置不当而失败, 在 `www_problem` 文件里就会出一个错误消息, 如下所示:

```
lynx: Can't access start file http://www.some-client-site.com
```

因为所有站点都使用 `makesite` 生成, 所以在站点配置里没有人为错误。

在使用这些种测试进行测试的同时, 也应通过 Web 浏览器测试这些站点。因为没有站点有实际内容。它们只是显示由 `makesite` 生成用于标识每个站点的 `index.html` 页面。

在上面所有的测试都成功以后, 该把内容从活动系统传输到新系统的时候了。

## 19.10 投入正常运行

只要文件被移入新试用区, 一些问题就可能暴露出来。例如, `XC News` 和它的客户的许多页面使用服务器端包含命令 (SSI) 运行显示通栏标题广告的脚本。虽然已对 Apache 服务器进行了正确的配置来支持 SSI 调用, 但是它并不显示这些广告; 相反, 在调用通栏标题广告脚本的每个页面上会出现错误消息。

出现这个问题是由于 Microsoft IIS 服务器运行 SSI 脚本的方式造成的。如下所示的 SSI 调用在基于 NT 的服务器上工作得很好, 但在 Apache 服务器上就不能工作:

```
<!--#exec cmd="perl.exe d:/var/www/xcnews/cgi-bin/rotate
_banner_ad.pl d:/var/www/xcnews/banners/giflist.txt" -->
```

像这样的 SSI 调用需要变为如下:

```
<!--#exec cmd="/www/xcnews/cgi-bin/rotate_banner_ad.pl
/www/xcnews/banners/giflist.txt" -->
```

称作 `rotate_banner_ad.pl` 的 perl 脚本也需要更改, 使它使用 `#!/path/to/perl` 行作为第一行, 但这在 NT 系统上并不需要。还需要检查任何对 `d:/var/www` 或任何其他 DOS 方式的路径信息的引用。一旦 SSI 调用变为标准的 SSI 调用, 并且脚本可以被 Apache 服务器运行, 脚本将正常工作。

对于使用 DOS 方式路径的 SSI 调用, 也会出现类似的问题。这些路径必须变成 UNIX 方式的路径。由于成百上千的文件需要修订, 让我们写一个简单的脚本去自动修复所有的文件, 脚本如列表 19-14 所示。

列表 19-14: `fixssi`

```
#!/bin/csh -f
```

```
#
# This script should be removed after AMINENS sites
# have completely moved away from NT (BVS)
# Kabir

set thisdir = '/bin/pwd';

echo "fixssi will translate NT SSI calls to Apache SSI calls."
echo "Current directory: $thisdir";
echo -n "Replacing perl.exe d:/var/www/htdocs/cgi-bin:"
/usr/bin/find . -type f -exec /usr/local/bin/fgres "perl.exe
d:/var/www/htdocs/cgi-bin " "/www/xcnews/cgi-bin" {} \ ;

echo "done."
```

当从/www 运行 fixssi 时，它将修复所有有 SSI 调用问题的站点。

**Tip** fgres 是可以从大多数 comp.unix.source 文档里获得的实用程序，也可通过 Web 搜索引擎找到。

应该将所有的 Perl CGI 脚本修改为使用#!/usr/bin/perl 行调用 Perl 解释程序。所有文件的权限位应这样设置：使每个 Web 服务器上的 httpd 进程能运行 CGI 脚本，写数据到 /www/cgi-bin 目录。

随着这些问题的解决，该是从旧系统转移到新系统的时候了。为了安全，运行另一轮测试，以确保 Web 站点设置正确和所有 CGI 脚本工作正常。这些测试显示：当从 HTML 表单收集数据时，一些 CGI 脚本发送 e-mail 数据到客户 e-mail 地址。这些需要进行修改，使它们使用 sendmail 邮递守护进程进行发送。在 CGI 脚本和其他内容文件进行最后一组测试后，一切显示正常，已准备好投入正常运行！

根据 InterNIC 的基于 Web 的 DNS 记录更新页，已经提交了改变所有有关域名的主、副域名服务器的请求。在短时间内，DNS 的改变已传播到 Internet 的各个角落。这中间确实有一段时间用户不能访问他们的 Web 站点，但你不能做任何事去帮助 Internet 的 DNS 传播。一些天以后，传播完成，所有站点正常运行在新系统上。

## 19.11 未来的可能性

使用循环 DNS 比不用好，但这不是最好的解决方案。不幸的，更好的方案比在自由 DNS 服务器软件上实现的循环 DNS 方案费用要高得多。使用循环 DNS 方案的潜在缺点有：

- ◆此方案不能检测服务器故障。因此，它将保持将客户的请求指向不正常工作的服务器，直至 DNS 由系统管理员手工修改为止。

◆当 DNS 记录改变时，不能被立即传播，这是因为 DNS 服务器为了减少带宽使用了缓冲。这种传播可以通过使用较小的 TTL 值来加快，但这将在网上形成大量与 DNS 有关传输。

为了改进这种方案可以做些什么呢？可以使用昂贵的负载平衡切换器，如 Cisco 的 AceDirector，它在网络级处理负载平衡。解决负载平衡的新的硬件解决方案正在出现以解决前面的问题。硬件方案通常运行在网络级，特殊切换硬件通过网络服务统计和其他相关信息检测负载。因为这种技术刚获得突破，所以价格很高。等待价格下降在将来使用硬件解决方案也许是个好主意。

## 19.12 小结

在本章中，讨论了如何创建由多个 Apache Web 服务器组成的 Web 网络。使用了实际的例子，这样能很好地了解出现在这样的项目中的细节。注意：实际项目确实包括服务器监视需求，由于篇幅的限制没在这里讨论它。

下一章，将讨论如何在 Windows 95 和 Windows NT 系统上运行 Apache。

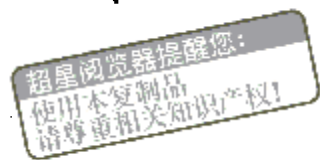
超星浏览器提醒您：  
使用本复制品  
请尊重相关知识产权！



## 第 20 章 用于 Windows 95/NT 的 Apache

本章内容:

- ◆ 将 Apache 移植到 Windows 所面对的问题
- ◆ 如何获取 Apache 的 Windows 版
- ◆ 在 Windows 上安装 Apache
- ◆ 在 Windows 上配置 Apache
- ◆ Windows 版专用的 Apache 命令



用于 Microsoft Windows 95/NT 的 Apache 是计划发行的下一个主要 Apache 版本。至少 Apache 团体是这样想的。发行这一主要版本是件大事, Apache 团体预计不可能很快实现。

一个 Apache 用户不能等待那么久, 他已把最新的 Apache beta 版移植到 Windows。这给 Apache 团体提供了新的选择: 现在他们拥有 Windows 平台的 Apache 的 beta 版。由于 Apache 团体的赞同, 可以从所有主要的 Apache 发布镜像站点(包括正式的 Apache Web 站点)获得这一 beta 版。

在本章中, 将讨论如何使用这一 beta 版(1.3b3 版), 预计对 Windows 有什么影响。必须注意, 一般认为 beta 版是不稳定的。除了试验以外不应用于任何其他目的。不要计划在它上面运行你的组织的 Web 站点, 如果你想知道 UNIX Web 服务器如何进入 Windows 市场, 可以阅读下一节。否则可以直接跳到第二节: 如何获得 Apache 的 Windows 版。

### 20.1 将 Apache 移植到 Windows 所面对的问题

移植任务经常是令人厌烦的, 它不适合于缺乏耐心的程序员。如果两个操作系统的主要方面根本不同, 如 UNIX 和 Windows, 移植任务就变得更为艰巨。

因为 Apache 是 UNIX 的产物(它是在 UNIX 平台上创建的), 它继承了所有 UNIX 软件设计的好的方面。Apache 使用了 pre-forking 服务器模型, 它几乎是 UNIX 界的实际标准。在这种模型里, 服务器的主进程只做很少的事情; 它创建(fork)子进程, 并由子进程去处理繁重的工作。这使得子进程的程序很复杂, 而主服务器的程序却很简单。

有人说 fork 吗? 在 Windows 的字典里并没有 fork 这个词汇。Windows 并不 fork 子进程。事实上, Windows 根本不创建子进程! Windows 服务器进程使用线程, 并被称为多线程服务器。线程不是独立的进程, 它是在服务器进程空间内运行的可执行代码, 它独立于其他线程完成一项任务。服务器进程被称为主线程, 而其他线程常被称为工作线程(这使人想到工蜂和蜂后之间的关系)。从移植 Apache 的角度看, 这些意味着什么呢?



因为在 Windows 上不能使用 fork, Apache 必须以单一服务器进程的方式运行, 服务器进程必须产生线程去服务请求。还有, 在工作线程接过请求之前主线程还需要做些工作。如果其中一个线程出现问题会发生什么情况呢? 将会发生灾难性后果——所有的线程都无法进行工作。对 Apache 讲, 这就意味着所有的 Web 服务都将停止。当然, 可以绕一个弯去解决这个问题。办法是启动另外一个 Apache 进程作为后备 Apache 服务器。当活动的 Web 服务器进程停止工作时, 后备服务器就替代它。用这种方法, 总的来说 Web 服务不会完全停止。

在开发多线程应用程序时, 另一个要求是数据对线程必须是安全的。一些 Apache 模块使用全局变量, 在 Apache 的多线程版中不能使用这些全局变量, 因为线程可能破坏数据。因此使用变量要特别小心, 必须确保每个线程有它自己的变量的拷贝。

除了多线程问题外, 另一个主要差别也会很快出现。Windows 通过外部动态连接库 (DLL) 文件加载模块, 而在 UNIX 里, 所有模块被编译成单个可执行文件。因此必须写一个特殊的模块, 叫作 `mod_dll`, 由它装入其他被编译成 DLL 文件的模块。

注意: 将 Apache 从 UNIX 移植到 Windows 可能遇到的问题不止这些, 还会有其他的问题。当前移植的版本并不会像 Windows 专用的程序一样工作, 因为它没有图形用户界面 (GUI), 仍是命令行形式的应用程序。这在以后的版本中将会得到改变。

现在让我们看看它有些什么。要记住它仅仅是个试验用的服务器。所以, 当它并不像你想像那样工作时, 不要见怪。

## 20.2 如何获取 Apache 的 Windows 版

在获取 Apache 的 Windows 版时, 有两种选择。可下载源代码发行版, 也可下载二进制文件的发行版。除非非常熟悉 Windows 下的 Visual C++ 或 Borland C++ 的开发过程, 否则不要下载源代码, 那是自寻烦恼。如果很想看 Apache 源代码, 可以只下载源代码。

竭力建议去获取最新发布的二进制版本。所发布的源代码和二进制文件都可从主要的 Apache Web 站点得到, 其站点的 URL 如下:

[www.apache.org](http://www.apache.org)

然而, 大概应当去最近的 Apache 镜像站点去下载软件。注意: 二进制发行版不包含任何文档。应当单独下载发布的文档。源代码和二进制文件也可从 CD-ROM 上获得。

## 20.3 在 Windows 上安装 Apache

安装过程是典型的基于 Windows 的软件安装。需要按如下步骤进行操作:

1. 运行从 Apache 镜像站点下载的自解压可执行文件。运行可执行文件时, 它会弹出如图 20-1 所示的对话框。

它显示了版权信息, 并告诉此软件包是用 Installshield 软件生成的。

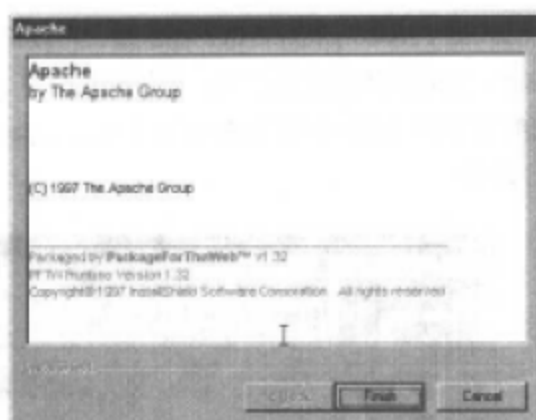


图 20-1 安装 Apache 的第一屏

2. 单击 Finish 按钮开始软件安装。将看到 Welcome 对话框, 如图 20-2 所示。

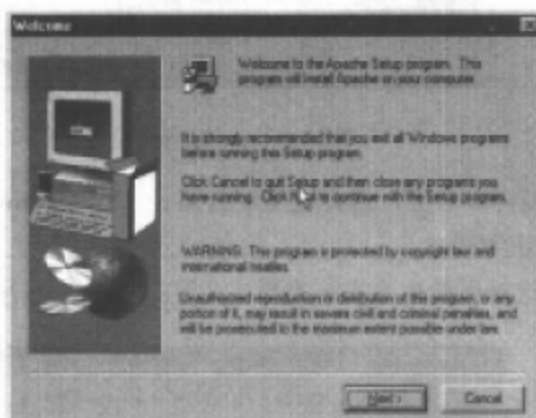


图 20-2 Welcome 屏

3. 单击 Next 按钮, 将看到 Apache 许可证条款, 如图 20-3 所示。如果同意这些条款, 单击 Yes 按钮。否则, 单击 No 按钮终止安装。

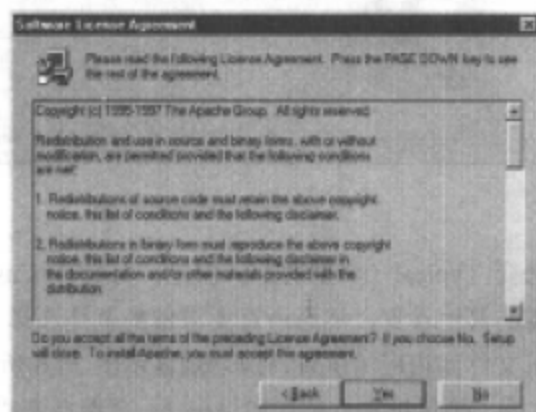


图 20-3 Apache 许可证屏

4. 如果同意, 将出现如图 20-4 所示对话框。在这里, 可以接受缺省的文件夹位置 (如 C:\Program Files\Apache) 也可选择其他文件夹。在缺省情况下, Apache 的 Windows 版在 \Apache\conf 下查找配置文件。因此, 如果将 Apache 安装在 \Apache 目录下, 在运行 Apache 时就不必指定配置文件路径。

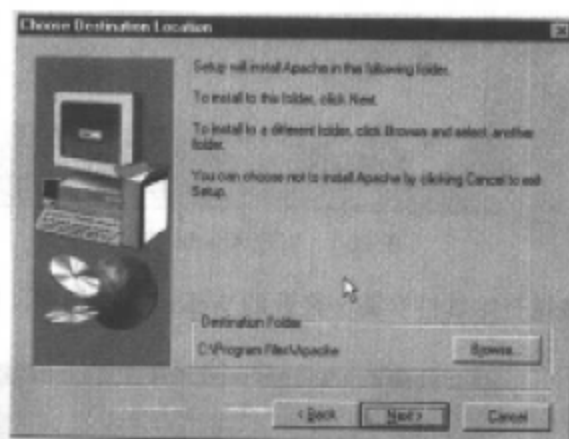


图 20-4 Apache 安装目录

然而, 为了与安装样本对应, 选择使用缺省路径 C:\Program Files\Apache。一旦选择了路径, 单击 Next 按钮继续。

5. 下一步将需要选择所希望的安装类型。此窗口如图 20-5 所示。

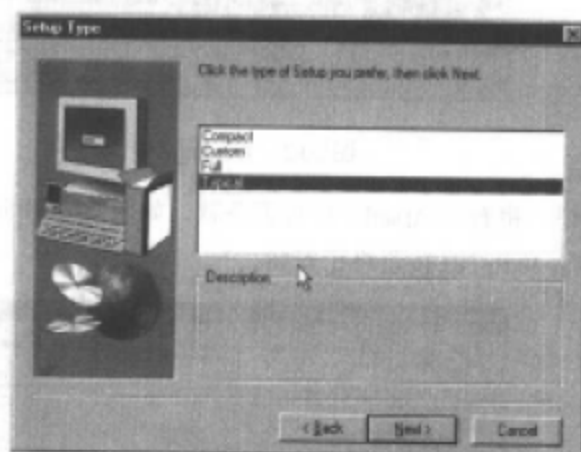


图 20-5 选择安装类型

如果想快一些, 选择 Typical (典型的)。但是, 可选择 Custom (自定义) 类型以便看一看安装过程中所需的其他设置。然而所有要求做的就是选择一个文件夹。选择安装类型后, 安装程序将拷贝文件到相应的位置, 成功则显示图 20-6 所示的窗口。

6. 如果决定安装文档, 确保把它安装在和 Apache 顶级目录相同的目录下。换句话说, 如果安装 Apache 到 C:\Program Files\Apache, 那么把文档安装在同一目录下。文档的安装程序将要求输入路径名, 如图 20-7 所示。



图 20-6 安装完成通知

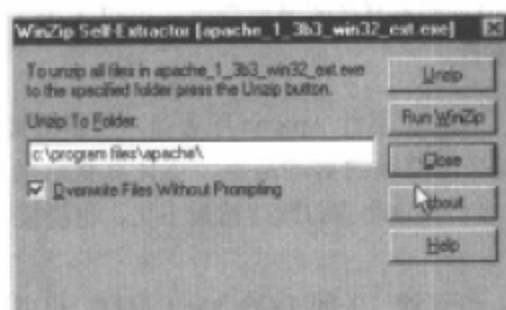


图 20-7 安装 Apache 的文档

现在，已经完成了 Apache 二进制文件的安装，并已为运行它作好了准备。当然，运行它的方式可能会因 Windows 版本不同而不同。

### 20.3.1 在 Windows 95 上运行 Apache

在 Windows 95 上，Apache 作为控制台应用程序运行。换句话说，或者从 Windows Explorer（不是 Internet Explorer）中单击它来运行，或者从如图 20-8 所示的 DOS 窗口中运行它。

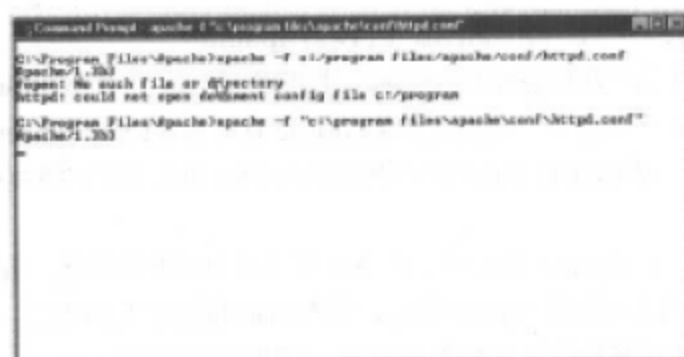


图 20-8 在 Windows 95 中运行 Apache

注意：当试图从 C:\Program Files\Apache 目录用如下的命令运行 Apache 时：

```
Apache -f c:/program files/apache/conf/httpd.conf
```

它不运行，当试用下面方式时：

```
Apache -f "c:\program files\apache\conf\httpd.conf"
```

它工作了。不同之处是在第二种方式里使用了双引号，因此 Apache 不把包含空格字符（在\program files\中）的路径当作两个命令行参数。还应注意到用 UNIX 分隔符代替 DOS 分隔符。然而，下面的行也工作：

```
Apache -f "C:/program files/apache/conf/httpd.conf"
```

因此，在命令行中两种分隔符都工作。然而用在配置文件里的所有配置路径信息必须使用 UNIX 分隔符。这些配置文件有 httpd.conf、access.conf 和 srm.conf。以控制台应用程序运行 Apache 时，会发现 Apache 除了显示 Apache 版本号外不显示任何其他信息。

**Tip** 也可以通过安装程序把服务器加到 Start 菜单中，从 Start 菜单中启动



Apache 服务器。

如果打算在计算机启动时就运行 Apache 服务器，可以把安装程序生成的 Start 菜单中的快捷启动图标拷贝到开始菜单的启动组里，当计算机启动时，会自动出现一个运行 Apache 服务器的 DOS 窗口。这个窗口既能被最小化，也能被删除（这是 DOS 窗口属性的选项）。

如果想停止服务器，既可通过关闭 DOS Shell 窗口，也可通过按 Ctrl+C 来实现。还没有停止 Apache 服务器更简单的方法。

### 20.3.2 在 Windows NT 上运行 Apache 作为一项服务

可以完全像在 Windows 95 中那样在 Windows NT 中运行 Apache。但在 Windows NT 4.0 或更高版本中也能运行 Apache 作为一项服务。为了作为 NT 一项服务运行 Apache，可以使用下面命令：

```
Apache -i
```

如果在其他目录而不是/Apache 目录中安装 Apache，则必须使用 -d 选项指定其所在目录。例如：

```
Apache -i -d "c:/program files/apache"
```

这将作为 NT 的一项服务安装 Apache，并告诉 Apache 在 C:/program files/apache 路径下寻找配置文件。一旦作为一项服务安装 Apache 就可从 NT Control Panel（控制面板）里控制如何运行它。从 Control Panel 窗口里选择 Apache 服务图标，将会看到如图 20-9 所示的窗口。

在缺省情况下，Apache 被配置为在 NT 启动时自动开始运行。当然，如果现在就想运行它，从图中列表中选择 Apache 服务，单击 Start 按钮手工启动它。

如果想从 NT 服务器服务中去掉 Apache，使用的命令如下：

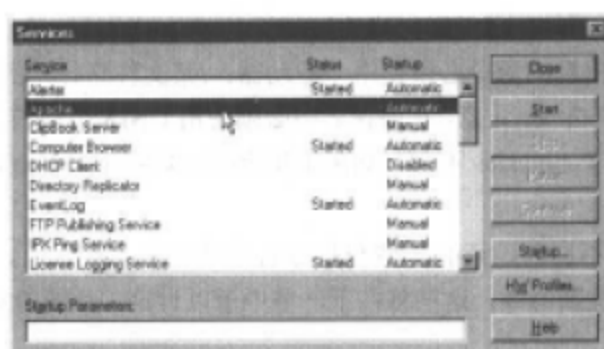


图 20-9 Apache 作为一项 NT 服务

`Apache -u -d "C:/program files/apache"`

这将从服务列表中删去 Apache，这个命令假定 Apache 的目录是 C:/Program Files /apache。如果必要，用你自己的 Apache 路径替换它。

一旦使 Apache 在 Windows 95 或 Windows NT 里运行后，应使用 Task Manager（任务管理程序）去辨别它是否真的运行了。图 20-10 显示了在 Task Manager 窗体里的 Apache 进程。

其中只有一个 Apache 是真正能够提供服务的。剩余的 Apache 进程是在当前进程死亡时才能启用的后备进程。可以控制启动多少个后备 Apache 进程，其使用的特殊配置选项将在下面予以讨论。



图 20-10 在 Task Manager 窗口中显示的 Apache 服务器进程

## 20.4 在 Windows 上配置 Apache

除了以下的一些改变外，Windows 的 Apache 配置同 UNIX 的配置（见第 2、4、5 章）

惊人的一致。

任何使用路径信息的命令必须使用“驱动器：/某文件路径”的格式代替 UNIX 下使用的“/某文件路径”格式。注意，在路径名中必须使用 UNIX 分隔符。例如，C:/www/mysite 是可以的，但 C:\www\mysite 不行。如果不指定驱动器字符，将假定为 Apache 所安装的驱动器。

除了路径信息不同外，还有一些与指令有关的改变。实际上，一些旧指令有了新含义，并且增加了一些新命令。这些将在下一节中予以讨论，从新的指令开始。

## 20.5 Windows 版专用的 Apache 命令

Apache 的 Windows 版本随带一个特殊模块，称为 mod\_dll。这个模块用于装入那些未被编译在 Apache 服务器内的外部模块。可以使用下面的命令：

```
Apache -l
```

将获得和 Apache 一起编译的模块列表。

在当前的 Apache 版本中编译在可执行文件中的模块如下：

- ◆http\_core.c
- ◆mod\_dll.c
- ◆mod\_mime.c
- ◆mod\_access.c
- ◆mod\_auth.c
- ◆mod\_negotiation.c
- ◆mod\_include.c
- ◆mod\_autoindex.c
- ◆mod\_dir.c
- ◆mod\_cgi.c
- ◆mod\_userdir.c
- ◆mod\_alias.c
- ◆mod\_env.c
- ◆mod\_log\_config.c
- ◆mod\_asis.c
- ◆mod\_imap.c
- ◆mod\_actions.c
- ◆mod\_setenvif.c
- ◆mod\_isapi.c

注意 mod\_dll 模块已经被建立在 Apache 可执行文件中了，否则 mod\_dll 将不能工作。同时，在 Apache 服务器根目录下的 modules 子目录中还可以发现其他模块。这些模块列表如下：

- ◆ApacheModuleAuthAnon.dll
- ◆ApacheModuleCERNMeta.dll
- ◆ApacheModuleDigest.dll
- ◆ApacheModuleExpires.dll
- ◆ApacheModuleHeaders.dll



- ◆ApacheModuleRewrite.dll
- ◆ApacheModuleStatus.dll
- ◆ApacheModuleUserTrack.dll

如果想在 Apache 服务器里使用上面的任一模块, 必须使用 `mod_dll nprovided` 命令, 这将在下一节说明。

除了新的 `mod_dll` 功能外, 还有一些命令的含义发生了变化。除了下面提到的命令外, 其他命令的工作方式不变。然而, 因为当前用的是 `beta` 版, 许多别的命令也许还不能正常工作。

改变含义的命令包括:

◆**StartServer**——这个命令指定后备服务器的数目。在当前服务器进程由于意外故障不能工作时, 它们之中的一个被启动以接替 Web 服务。如果启动很多后备进程, 大概除浪费内存外并不能起其他作用。推荐使用缺省值。

◆**MinSpareServers**——因为真正使用的只有一个服务器(其他都是后备的), 在 Windows 里不需使用此命令。

◆**MaxSpareServer**——因为真正使用的只有一个服务器(其他都是后备的), 在 Windows 里不需要使用此命令。

◆**MaxRequestPerChild**——在 Windows 下, 只用一个进程处理所有的请求。所以在达到此命令指定的最大请求数时, 没有子进程被删除。如果设置一个较小的数, 服务器将退出。但是, 如果设为 0, 服务将根本不退出。

### 20.5.1 LoadModule

语 法: `LoadModule 模块名 到模块的路径`

使用范围: 服务器配置

这个命令允许在 Apache 服务器启动时装入 DLL 文件作为 Apache 模块。例如:

```
LoadModule mod_digest modules/ApacheModuleDigest.dll
```

这将装入 `modules/ApacheModuleDigest.dll` DLL 文件作为 `mod_digest` 模块。如果到模块文件的路径没有 `/` 前缀, 则假定是服务器根目录的子目录。

### 20.5.2 LoadFile

语 法: `LoadFile 文件名 到 DLL 文件的路径`

使用范围: 服务器配置

这个命令允许装入 DLL 模块自身所需要的 DLL 文件, 如果到文件的路径没有 `/` 前缀, 则假定是服务器根目录的子目录。

### 20.5.3 ThreadsPerChild

语 法: `ThreadPerChild 线程数`

缺 省 值: `ThreadPerChild 50`

使用范围: 服务器配置

超星浏览器提醒您:  
使用本复制品  
请尊重版权!

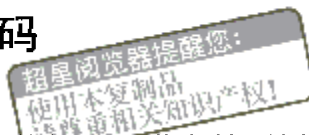
这个称为 ThreadPerChild 的新的核心命令设置 Apache 服务器可使用的线程数目。因为每个线程都能处理一个请求，它等效于限制服务器能服务的请求总数。

## 20.6 小 结

在本章中，讨论了 Apache 的 Windows 版的当前情况。当满怀希望地得知 Apache 团体正在做许多基础工作以确保 Apache 在所有平台上的领先地位时，你就会受到鼓舞。当然，为什么所有好东西只能存在于 UNIX 上呢？

超星浏览器提醒您  
使用本复制品  
请尊重相关知识产权!

## 附录 A HTTP/1.1 状态码



对于来自 Web 客户的每个请求，Web 服务器都必须给客户返回 HTTP 状态码。该状态码是由三位数字组成的编号。Web 客户通过察看在 HTTP 状态行头标中发送的状态码，就能够尝试了解服务器的响应。与此状态码一起的还有一条短语，叫做原因短语，它的目的是向用户提供简要的解释。例如，HTTP 状态行头标看起来可能像下面那样：

HTTP/1.1 404 Not Found

在这里，“404”是状态码，而“Not Found”则是原因短语。在典型的 Web 客户端，Web 浏览器会在浏览器窗口中显示该“Not Found”短语。在最新的 HTTP/1.1 规范中，有五类不同的状态码。这些类别将在下面进行探讨。

### A.1 信息状态码（100~199）

此种类型状态码的目的是让客户端知道服务器正处于满足请求的过程中。这些状态码只是通知性质的，用户不必对它们中的任何一条采取行动。注意，HTTP/1.0 中没有定义 1xx 状态码，因此 1xx 状态码一定不能被发送给 HTTP/1.0 的客户。当前定义的 1xx 状态码如下所示：

100 Continue	服务器发送此状态码，让客户知道服务器已经准备好接收请求的其余部分。
101 Switching Protocols	当服务器愿意将应用协议切换到由客户在 Upgrade（升级）请求头标中指定的协议时，服务器发送此状态码。只有在新协议比已有的协议有优势时，才应该进行切换。例如，客户可能请求服务器使用比当前正在使用的 HTTP 协议更新的 HTTP 协议。在这种情况下，如果有可能，服务器应该进行切换。

### A.2 客户请求成功（200~299）

如果服务器返回的状态码在 200~299 范围内，则客户的请求被成功地接收并接受了。当前已定义的 2xx 状态码如下所示：

200 OK	服务器已经成功地处理了该请求，并附上了所请求的文档。
201 Created	服务器已经成功地创建了 Location 头标中指定的新的 URI。
202 Accepted	处理请求已经被接受了，但服务器尚未完成对请求的处理。

203 Non-Authoritative Information	在响应头标中的元信息不是来自该服务器，而是从另一个服务器上复制过来的。
204 No Content	请求是完备的，但没有新的信息需要发送回去。客户端应该继续显示当前的文档。
205 Reset Content	客户端应该复位当前的文档。当需要复位 HTML 表单以清除所有输入字段中已有的值时，这是有用的。
206 Partial Content	服务器已经满足了对资源的部分 GET 请求。此状态码被用来响应 Range 请求。服务器发送 Content-Range (内容范围) 头标来指示附上了哪一段数据。

### A.3 对请求进行了重新定向 (300~399)

服务器发送在 300~399 范围内的状态码给客户，使客户知道它需要完成进一步的操作以完成该请求。当前定义的 3xx 状态码如下所示：

300 Multiple Choices	所请求的资源对应于一组文档。服务器可以发送每个文档和它的具体位置的信息以及内容协商信息，允许客户选择其中的一个。
301 Moved Permanently	所请求的资源不在服务器上。服务器发送 Location 头标，将客户重新定向到新的 URI。客户在以后将所有的请求定向到新 URI。
302 Moved Temporarily	所请求的资源已经暂时移走了。服务器发送 Location 头标，将客户重定向到新的 URI。客户在以后的请求中继续使用老的 URI。
303 See Other	所请求的资源可以在由 Location 头标指示的不同位置处找到，而且客户应该使用 GET 方法来获取它。
304 Not Modified	服务器用此状态码来响应 If-Modified-Since 请求头标。它指示自指定日期以来所请求的文档尚未被修改，而且客户应该使用它缓存的拷贝。
305 Use Proxy	客户应该使用在 Location 头标中指定的代理来获取所请求的资源。

### A.4 客户请求不完备 (400~499)

服务器发送在 400~499 范围内的状态码，指示客户的请求不完备，需要更多的信息来完成资源请求。当前定义的 4xx 状态码如下所示：

400 Bad Request	服务器在客户请求中检测到语法错误。
401 Unauthorized	这一请求需要进行用户认证。服务器发送 www-

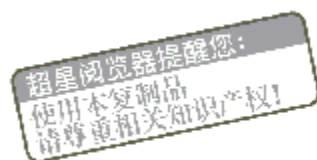
	Authenticate 头标来指示所请求资源的认证类型和区域。
402 Payment Required	此状态码保留为将来使用。
403 Forbidden	对所请求资源的访问被禁止。客户不应该重复该请求。
404 Not Found	所请求的文档不在服务器上。
405 Method Not Allowed	客户所使用的请求方法不可接受。服务器发送 Allow 头标, 说明用什么方法访问所请求的资源是可以接受的。
406 Not Acceptable	根据服务器接收到的 accept 头标, 请求的资源不存在客户可以接受的格式。如果该请求不是一个 HEAD 请求, 则服务器可以发送 Content-Language 头标、Content-Encoding 头标和 Content-Type 头标来指示哪些格式是可用的。
407 Proxy Authentication Required	对代理服务器的未经授权的访问请求。客户必须先由代理进行认证。服务器发送 Proxy-Authenticate 头标, 指示对所有请求资源的验证方案和区域。
408 Request Time-Out	客户不能在服务器所使用的请求超时时间段内完成它的请求。然而, 客户可以重复该请求。
409 Conflict	客户请求与另一个请求冲突。服务器能够与状态码一起发送关于冲突类型的信息。
410 Gone	请求的资源在服务器上被永久地移走了。
411 Length Required	客户必须在它的请求中提供 Content-Length 头标。
412 Precondition Failed	当客户发送带有一个或多个 If...头标的请求时, 服务器使用此状态码来指示在这些头标中指定条件中的一个或多个是假的。
413 Request Entity Too Large	服务器拒绝处理该请求, 因为它的消息体太大了。服务器可以关闭该连接, 以停止客户继续该请求。
414 Request-URI Too Long	服务器拒绝处理该请求, 因为所指定的 URI 太长了。
415 Unsupported Media Type	服务器拒绝处理该请求, 因为它不支持消息体的格式。

## A.5 服务器错误 (500~599)

当服务器发生错误而且不能完成请求时, 返回在 500~599 范围内的状态码。当前定义的 5xx 状态码如下所示:


500 Internal Server Error	服务器配置设置或外部程序导致了错误。
501 Not Implemented	服务器不支持为完成该请求所要求的功能。
502 Bad Gateway	服务器从上端服务器或代理处遇到非法响应。

503 Service Unavailable	该项服务暂时不可使用。服务器可以发送 <b>Retry-After</b> 头标，以指示该项服务何时可以再次可用。
504 Gateway Time-out	网关或代理超时了。
505 HTTP Version Not Supported	不支持客户使用的 <b>HTTP</b> 版本。









\E	结束大小写修改
\u	将下一个字符改为大写
\Q	引用直到\E 为止的元字符

如果需要在规则表达式中将元字符用作正常字符，可以使用 `\metachar` (\元字符) 的格式来取消它的特殊意义。这种情况的一个例子是 `\$`，它是一个正常美元符号。在规则表达式中所使用的标准量词如下所示：

*	匹配 0 次或多次
+	匹配 1 次或多次
?	匹配 1 次或 0 次
{n}	正好匹配 n 次
{n,}	最少匹配 n 次
{n,m}	最少匹配 n 次，但最多匹配 m 次

还要注意，一个 `|` 字符是作为 OR 操作符来对待的。一对小括号 `()` 使你能够在规则表达式中组合字符组。一对方括号 `[]` 则创建了字符类或范围。

让我们重新看一下最开始的那个例子：

```
([a-z]+)\.([a-z]+)\.([a-z]+)
```

正如前面所提到的，此表达式可以用来匹配诸如 `www.idgbooks.com` 这样的字符串。第一个 `[a-z]+` 指定在 `a` 到 `z` 范围内的一个或多个字符来匹配由第一对小括号所指定的字符组。如果发现了一个匹配，那么就可以用 `$1` 来访问所匹配的内容。在此表达式中有三对小括号。第一对小括号（从左边数）是 `$1`，第二对是 `$2`，而第三对则是 `$3`。注意，`\` 被用来改变在组之间的点 `(.)` 元字符的意义。

下面是另外两个例子：

```
◆ ^foo\.htm$
```

它将匹配一个字符串 `foo.htm`。它不会匹配 `afoo.htm`，`^` 元字符被用来指定匹配字符串必须以字符 `f` 开头。它也不会匹配 `foo.html`，因为 `$` 元字符被用来指定匹配字符串必须以字符 `m` 结尾。

```
◆ ^www\.([^.]+)\.host\.com(.*)
```

它将匹配一个字符串，如 `www.username.host.com STATUS=java`，并且 `$1` 会被赋值为 `username`，而 `$2` 会保存跟在字符串 `www.username.host.com` 之后的任何字符。在本例中，`$2` 将保存 `STATUS=java`。

## 附录 C 可用于 Apache 的 Internet 资源



在本附录中，读者会看到 Web 站点、Usenet 新闻组以及邮递表地址的列表，你可能发现它们对你很有用处。

### C.1 自由资源

下面是一些用于 Apache 的自由 Internet 资源

#### C.1.1 Web 站点

Official Apache Web site	<a href="http://www.apache.org">www.apache.org</a>
Apache Module Registry	<a href="http://modules.apache.org/">http://modules.apache.org/</a>
ApacheSSL	<a href="http://www.apache-ssl.org/">http://www.apache-ssl.org/</a>
Apache/Perl Integration Project	<a href="http://perl.apache.org/">http://perl.apache.org/</a>
Apache GUI Project	<a href="http://butler.disa.mil/ApacheConfig/">http://butler.disa.mil/ApacheConfig/</a>
Java-Apache Project	<a href="http://jave.apache.org/">http://jave.apache.org/</a>
Apache for OS/2	<a href="http://www.slink.com/ApacheOS2/">http://www.slink.com/ApacheOS2/</a>
Apache for Amiga	<a href="http://www.xs4all.nl/~albertv/apache/">http://www.xs4all.nl/~albertv/apache/</a>

### C.2 Usenet 新闻组

还可以在 15 个不同的新闻组中找到有关 World Wide Web 主题的信息，对它们进行了划分。请使用与你的主题关系最密切的新闻组。在邮递任何内容之前，请一定阅读该新闻组的 FAQ，并确保邮递到正确的新闻组。

#### C.2.1 与 Web 服务器有关的新闻组

`comp.infosystems.www.servers.unix`

此新闻组讨论用于 UNIX 平台的 Web 服务器。可能的主题有：配置问题/解决方案、安全问题、目录结构以及隐错报告。

`comp.infosystems.www.servers.ms-windows`

此新闻组包括用于 MS Windows 和 NT 平台的 Web 服务器。可能的主题有：配置问题/解决方案、安全问题、目录结构以及隐错报告。

`comp.infosystems.www.servers.mac`

此新闻组包含用于 Macintosh (MacOS) 平台的 Web 服务器。可能的主题有：配置问

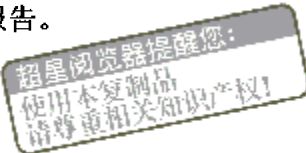
题/解决方案、安全问题、目录结构以及隐错报告。

`comp.infosystems.www.servers.misc`

此新闻组讨论用于其他平台的 Web 服务器, 如 Amiga、VMS 等等。可能的主题有: 配置问题/解决方案、安全问题、目录结构以及隐错报告。

`japan.www.servers.apache`

此新闻组包括在日本进行的有关 Apache 的讨论。



### C.2.2 与创作 Web 页有关的新闻组

`comp.infosystems.www.authoring.cgi`

此新闻组讨论有关通用网关接口 (CGI) 脚本的开发。可能的主题有: 如何处理表单的结果, 如何在运行中生成图像, 以及如何把其他以交互方式取得的结果放在一起。

`comp.infosystems.www.authoring.html`

此新闻组讨论超文本标识语言 (HTML) 用于 Web 页创作有关的问题。可能的主题有: HTML 编辑器、格式化技巧, 以及当前的和提议中的 HTML 标准。

`comp.infosystems.www.authoring.images`

此新闻组讨论用于 Web 页的图像的创作和编辑。可能的主题有: 如何更好利用 Web 的图像显示能力, 以及建立图像映射的共同问题和解决方案。

`comp.infosystems.www.authoring.misc`

此新闻组覆盖了其他的 `comp.infosystems.www.authoring.*` 组没有讨论的各种 Web 页的创作问题。可能的主题有: 声音和电视的使用以及诸如此类的其他内容。

### C.2.3 与 Web 浏览器有关的新闻组

`comp.infosystems.www.browsers.ms-windows`

此新闻组讨论用于 MS Windows 和 NT 平台的 Web 浏览器。可能的主题有: 配置问题/解决方案、外部浏览器 (helper 应用程序) 以及隐错报告。

`comp.infosystems.www.browsers.mac`

此新闻组讨论用于 Macintosh 平台的 Web 浏览器。可能的主题有: 配置问题/解决方案、外部浏览器以及隐错报告。

`comp.infosystems.www.browsers.x`

此新闻组讨论用于 X-Windows 系统的 Web 浏览器。可能的主题有: 配置问题/解决方案、外部浏览器 (helper 应用程序) 以及隐错报告。

`comp.infosystems.www.browsers.misc`

此新闻组讨论用于所有其他平台的 Web 浏览器。可能的主题有: 配置问题/解决方案、外部浏览器 (helper 应用程序) 以及隐错报告。所包括的平台有: Amiga、DOS、VMS 以及 UNIX 文本模式。