# Apache日志介绍

想要知道什么人在什么时候浏览了网站的哪些内容吗？查看Apache的访问日志就可以知道。

## 1、什么是Apache日志

Apache内建了记录服务器活动的功能，这就是它的日志功能。这个《Apache日志》系列文章介绍的就是Apache的访问日志、错误日志，以及如何分析日志数据，如何定制Apache日志，如何从日志数据生成统计报表等内容。

如果Apache的安装方式是默认安装，服务器一运行就会有两个日志文件生成。这两个文件是access\_log（在Windows上是access.log）和error\_log（在Windows上是error.log）。采用默认安装方式时，这些文件可以在/usr/local/apache/logs下找到；对于Windows系统，这些日志文件将保存在Apache安装目录的logs子目录。不同的包管理器会把日志文件放到各种不同的位置，所以你可能需要找找其他的地方，或者通过配置文件查看这些日志文件配置到了什么地方。

|  |
| --- |
| 总结：访问日志和错误日志的位置和名字  Windows: <Apache安装目录>\logs\access.log | error.log      Linux:  /usr/local/apache/logs/access\_log | error\_log |
| 访问日志：  访问服务器的远程机器的地址：可以得知浏览者来自何方  浏览者访问的资源：可以得知网站中的哪些部分最受欢迎  浏览者的浏览时间：可以从浏览时间（如工作时间或休闲时间）对网站内容进行调整  浏览者使用的浏览器：可以根据大多数浏览者使用的浏览器对站点进行优化  错误日志：  获知失效链接  获知 CGI 错误  获知用户认证错误 |

## 2、如何分析Apache日志

（学习SEO（搜索引擎优化）不看日志是不行的,或者说不看日志你就不能成为一个真正的seo高手,经常的注意一下自己网站的日志分析蜘蛛爬行的规律是很有用的,另外日志还有助于解决各种安全问题,如你不知道的IP在访问你的后台等.）

　　日志根据服务器的不同而不同,一般有IIS和Apache两种,下面我们依次来看一下.

　　第一步自然是得到日志文件，这个各个站都是不同的,有的是isp提供下载只能保存几天的,也有的是从ftp可以找到日志,具体要询问自己的空间提供商,如果是你自己的服务器的话, IIS的日志可以在Windows文件夹下的system32\LogFiles\W3SVC1下.而Apache的日志默认是在安装地址的logs文件夹下。

## 3、Apache返回代码含义

### 3.1 访问日志代码含义

"100" : Continue 客户必须继续发出请求

　　"101" : witching Protocols 客户要求服务器根据请求转换HTTP协议版本　200交易成功

　　"200" : OK 交易成功

　　"201" : Created 提示知道新文件的URL

　　"202" : Accepted 接受和处理、但处理未完成

　　"203" : Non-Authoritative Information 返回信息不确定或不完整

　　"204" : No Content 请求收到，但返回信息为空

　　"205" : Reset Content 服务器完成了请求，用户代理必须复位当前已经浏览过的文件

　　"206" : Partial Content 服务器已经完成了部分用户的GET请求

　　"300" : Multiple Choices 请求的资源可在多处得到

　　"301" : Moved permanently 删除请求数据

　　"302" : Found 在其他地址发现了请求数据

　　"303" : See Other 建议客户访问其他URL或访问方式

　　"304" : Not Modified 客户端已经执行了GET，但文件未变化

　　"305" : Use Proxy 请求的资源必须从服务器指定的地址得到

　　"306" 前一版本HTTP中使用的代码，现行版本中不再使用

　　"307" : Temporary Redirect 申明请求的资源临时性删除

　　"400" : Bad Request 错误请求，如语法错误

　　"401" : Unauthorized 请求授权失败

　　"402" : Payment Required 保留有效ChargeTo头响应

　　"403" : Forbidden 请求不答应

　　"404" : Not Found 没有发现文件、查询或URl

　　"405" : Method Not Allowed 用户在Request-Line字段定义的方法不答应

　　"406" : Not Acceptable 根据用户发送的Accept拖，请求资源不可访问

　　"407" : Proxy Authentication Required 类似401，用户必须首先在代理服务器上得到授权

　　"408" : Request Time-out 客户端没有在用户指定的饿时间内完成请求

　　"409" : Conflict 对当前资源状态，请求不能完成

　　"410" : Gone 服务器上不再有此资源且无进一步的参考地址

　　"411" : Length Required 服务器拒绝用户定义的Content-Length属性请求

　　"412" : precondition Failed 一个或多个请求头字段在当前请求中错误

　　"413" : Request Entity Too Large 请求的资源大于服务器答应的大小

　　"414" : Request-URI Too Large 请求的资源URL长于服务器答应的长度

　　"415" : unsupported Media Type 请求资源不支持请求项目格式

　　"416" : Requested range not satisfiable请求中包含Range请求头字段，在当前请求资源范围内没有range指示值，请求也不包含If-Range请求头字段

　　"417" : Expectation Failed 服务器不满足请求Expect头字段指定的期望值，假如是代理服务器，

　　"500" : Internal Server Error 服务器产生内部错误

　　"501" : Not Implemented 服务器不支持请求的函数

　　"502" : Bad Gateway 服务器暂时不可用，有时是为了防止发生系统过载

　　"503" : Service Unavailable 服务器过载或暂停维修

　　"504" : Gateway Time-out 关口过载，服务器使用另一个关口或服务来响应用户，等待时间设定值较长

"505" : HTTP Version not supported 服务器不支持或拒绝支请求头中指定的HTTP版本

### 3.2 错误日志级别含义

|  |  |  |
| --- | --- | --- |
| Level | Description | Example |
| emerg | Emergencies - system is unusable. | "Child cannot open lock file. Exiting" |
| alert | Action must be taken immediately. | "getpwuid: couldn't determine user name from uid" |
| crit | Critical Conditions. | "socket: Failed to get a socket, exiting child" |
| error | Error conditions. | "Premature end of script headers" |
| warn | Warning conditions. | "child process 1234 did not exit, sending another SIGHUP" |
| notice | Normal but significant condition. | "httpd: caught SIGBUS, attempting to dump core in ..." |
| info | Informational. | "Server seems busy, (you may need to increase StartServers, or Min/MaxSpareServers)..." |
| debug | Debug-level messages | "Opening config file ..." |
| trace1 | Trace messages | "proxy: FTP: control connection complete" |
| trace2 | Trace messages | "proxy: CONNECT: sending the CONNECT request to the remote proxy" |
| trace3 | Trace messages | "openssl: Handshake: start" |
| trace4 | Trace messages | "read from buffered SSL brigade, mode 0, 17 bytes" |
| trace5 | Trace messages | "map lookup FAILED: map=rewritemap key=keyname" |
| trace6 | Trace messages | "cache lookup FAILED, forcing new map lookup" |
| trace7 | Trace messages, dumping large amounts of data | "| 0000: 02 23 44 30 13 40 ac 34 df 3d bf 9a 19 49 39 15 |" |
| trace8 | Trace messages, dumping large amounts of data | "| 0000: 02 23 44 30 13 40 ac 34 df 3d bf 9a 19 49 39 15 |" |

## 4、apache日志详解

### 4.1 访问日志access\_log

正如其名字所示，访问日志access\_log记录了所有对Web服务器的访问活动。下面是访问日志中一个典型的记录：

216.35.116.91 - - [19/Aug/2000:14:47:37 -0400] “GET / HTTP/1.0″ 200 654

这行内容由7项构成，上面的例子中有两项空白，但整行内容仍旧分成了7项。

第一项信息是远程主机的地址，即它表明访问网站的究竟是谁。在上面的例子中，访问网站的主机是216.35.116.91。随便说一句，这 个地址属于一台名为si3001.inktomi.com的机器（要找出这个信息，可以使用nslookup工具查找DNS），inktomi.com是 一家制作Web搜索软件的公司。可以看出，仅仅从日志记录的第一项出发，我们就可以得到有关访问者的不少信息。

（默认情况下，第一项信息只是远程主机的IP地址，但我们可以要求Apache查出所有的主机名字，并在日志文件中用主机名字来替代IP地 址。然而，这种做法通常不值得推荐，因为它将极大地影响服务器记录日志的速度，从而也就减低了整个网站的效率。另外，有许多工具能够将日志文件中的IP地 址转换成主机名字，因此要求Apache记录主机名字替代IP地址是得不偿失的。

然而，如果确实有必要让Apache找出远程主机的名字，那么我们可以使用如下指令：HostNameLookups on

如果HostNameLookups设置成double而不是on，日志记录程序将对它找到的主机名字进行反向查找，验证该主机名字确实指向了原来出现的IP地址。默认情况下HostNameLookups设置为off。）

上例日志记录中的第二项是空白，用一个“-”占位符替代。实际上绝大多数时候这一项都是如此。这个位置用于记录浏览者的标识，这不只是浏览者的登录名字，而是浏览者的email地址或者其他唯一标识符。这个信息由identd返回，或者直接由浏览器返回。很早的时候，那时 Netscape 0.9还占据着统治地位，这个位置往往记录着浏览者的email地址。然而，由于有人用它来收集邮件地址和发送垃圾邮件，所以它未能保留多久，很久之前市场上几乎所有的浏览器就取消了这项功能。因此，到了今天，我们在日志记录的第二项看到email地址的机会已经微乎其微了。

日志记录的第三项也是空白。这个位置用于记录浏览者进行身份验证时提供的名字。当然，如果网站的某些内容要求用户进行身份验证，那么这项信息是不会空白的。但是，对于大多数网站来说，日志文件的大多数记录中这一项仍旧是空白的。

日志记录的第四项是请求的时间。这个信息用方括号包围，而且采用所谓的“公共日志格式”或“标准英文格式”。因此，上例日志记录表示请求的时间是2000年8月19日星期三14:47:37。时间信息最后的“-0400”表示服务器所处时区位于UTC之前的4小时。

日志记录的第五项信息或许是整个日志记录中最有用的信息，它告诉我们服务器收到的是一个什么样的请求。该项信息的典型格式是“METHOD RESOURCE PROTOCOL”，即“方法 资源 协议”。在上例中，METHOD是GET，其他经常可能出现的METHOD还有POST和HEAD。此外还有不少可能出现的合法METHOD，但主要就是这三种。

RESOURCE是指浏览者向服务器请求的文档，或URL。在这个例子中，浏览者请求的是“/”，即网站的主页或根。大多数情况下，“/”指向DocumentRoot目录的index.html文档，但根据服务器配置的不同它也可能指向其他文件。

PROTOCOL通常是HTTP，后面再加上版本号。版本号或者是1.0，或者是1.1，但出现1.0的时候比较多。我们知道，HTTP协议是Web得以工作的基础，HTTP/1.0是HTTP协议的早期版本，而1.1是最近的版本。当前大多数Web客户程序仍使用1.0版本的HTTP协议。

日志记录的第六项信息是状态代码，详细的状态代码请查看第3节。它告诉我们请求是否成功，或者遇到了什么样的错误。大多数时候，这项值是200，它表示服务器已经成功地响应浏览器的请求，一切正常。但一般地说，以2开头的状态代码表示成功，以3开头的状态代码表示由于各种不同的原因用户请求被重定向到了其他位置，以4开头的状态代码表示客户端存在某种错误，以5开头的状态代码表示服务器遇到了某个错误。

日志记录的第七项表示发送给客户端的总字节数。它告诉我们传输是否被打断（即，该数值是否和文件的大小相同）。把日志记录中的这些值加起来就可以得知服务器在一天、一周或者一月内发送了多少数据。

|  |
| --- |
| **总结：日志名字段所代表的内容：**  **1、远程主机 2、空白（email） 3、空白（登录名） 4、请求时间 5、方法＋资源＋协议 6、状态代码 7、发送字节数** |
| 【总结】       方法METHOD: GET、POST、HEAD、……       资源RESOURCE: /、index.html、/default/index.php、……（请求的文件）      协议 PROTOCOL: HTTP+版本号 |

### 4.1 错误日志error\_log

错误日志和访问日志一样也是Apache的标准日志。本文分析错误日志的内容，介绍如何设置和错误日志相关的选项，文档错误和CGI错误的分类，以及如何方便地查看日志内容，等等。

一、位置和内容

前文讨论了Apache的访问日志，包括它的内容、格式和如何设置访问日志有关的选项。本文我们要讨论的是另外一种Apache标准日志——错误日志。

错误日志无论在格式上还是在内容上都和访问日志不同。然而，错误日志和访问日志一样也提供丰富的信息，我们可以利用这些信息分析服务器的运行情况、哪里出现了问题。

错误日志的文件名字是error\_log，但如果是Windows平台，则错误日志的文件名字是error.log。错误日志的位置可以通过ErrorLog指令设置：

ErrorLog logs/error.log

除非文件位置用“/”开头，否则这个文件位置是相对于ServerRoot目录的相对路径。如果Apache采用默认安装方式安装，那么错 误日志的位置应该在/usr/local/apache/logs下。但是，如果Apache用某种包管理器安装，错误日志很可能在其他位置。

正如其名字所示，错误日志记录了服务器运行期间遇到的各种错误，以及一些普通的诊断信息，比如服务器何时启动、何时关闭等。

我们可以设置日志文件记录信息级别的高低，控制日志文件记录信息的数量和类型。这是通过LogLevel指令设置的，该指令默认设置的级别 是error，即记录称得上错误的事件。有关该指令中允许设置的各种选项的完整清单，请参见<http://httpd.apache.org/docs/current/mod/core.html#loglevel> 的Apache文档。

大多数情况下，我们在日志文件中见到的内容分属两类：文档错误和CGI错误。但是，错误日志中偶尔也会出现配置错误，另外还有前面提到的服务器启动和关闭信息。

二、文档错误

文档错误和服务器应答中的400系列代码相对应，最常见的就是404错误——Document Not Found（文档没有找到）。除了404错误以外，用户身份验证错误也是一种常见的错误。

404错误在用户请求的资源（即URL）不存在时出现，它可能是由于用户输入的URL错误，或者由于服务器上原来存在的文档因故被删除或移动。（顺便说一下，按照Jakob Nielson的意见，在不提供重定向或者其他补救措施的情况下，我们永远不应该移动或者删除Web网站的任何资源。Nielson的更多文章，请参见http://www.zdnet.com/devhead/alertbox/。）

当用户不能打开服务器上的文档时，错误日志中出现的记录如下所示：

[Fri Aug 18 22:36:26 2000] [error]

[client 192.168.1.6] File does not exist:

/usr/local/apache/bugletdocs/Img/south-korea.gif

可以看到，正如访问日志access\_log文件一样，错误日志记录也分成多个项。

错误记录的第一项是日期/时间标记，注意它们的格式和access\_log中日期/时间的格式不同。access\_log中的格式被称为“标准英文格式”，这或许是历史跟我们开的一个玩笑，但现在要改变它已经太迟了。

错误记录的第二项是当前记录的级别，它表明了问题的严重程度。这个级别信息可能是LogLevel指令的文档中所列出的任一级别（参见前面 LogLevel的链接），error级别处于warn级别和crit级别之间。404属于error错误级别，这个级别表示确实遇到了问题，但服务器还 可以运行。

错误记录的第三项表示用户发出请求时所用的IP地址。

记录的最后一项才是真正的错误信息。对于404错误，它还给出了完整路径指示服务器试图访问的文件。当我们料想某个文件应该在目标位置却出 现了404错误时，这个信息是非常有用的。此时产生这种错误的原因往往是由于服务器配置错误、文件实际所处的虚拟主机和我们料想的不同，或者其他一些意料 不到的情况。

由于用户身份验证问题而出现的错误记录如下所示：

[Tue Apr 11 22:13:21 2000]

[error] [client 192.168.1.3] user rbowen@rcbowen.

com: authentication failure for “/cgi-bin/hirecareers/company.cgi”:

password mismatch

注意，由于文档错误是用户请求的直接结果，因此它们在访问日志中也会有相应的记录。

三、CGI错误

错误日志最主要的用途或许是诊断行为异常的CGI程序。为了进一步分析和处理方便，CGI程序输出到 STDERR（Standard Error，标准错误设备）的所有内容都将直接进入错误日志。这意味着，任何编写良好的CGI程序，如果出现了问题，错 误日志就会告诉我们有关问题的详细信息。

然而，把CGI程序错误输出到错误日志也有它的缺点，错误日志中将出现许多没有标准格式的内容，这使得用错误日志自动分析程序从中分析出有用的信息变得相当困难。

下面是一个例子，它是调试Perl CGI代码时，错误日志中出现的一个错误记录：

[Wed Jun 14 16:16:37 2000] [error] [client 192.168.1.3] Premature

end of script headers: /usr/local/apache/cgi-bin/HyperCalPro/announcement.cgi

Global symbol “$rv” requires explicit package name at

/usr/local/apache/cgi-bin/HyperCalPro/announcement.cgi line 81.

Global symbol “%details” requires explicit package name at

/usr/local/apache/cgi-bin/HyperCalPro/announcement.cgi line 84.

Global symbol “$Config” requires explicit package name at

/usr/local/apache/cgi-bin/HyperCalPro/announcement.cgi line 133.

Execution of /usr/local/apache/cgi-bin/HyperCalPro/announcement.cgi

aborted due to compilation errors.

可以看到，CGI错误和前面的404错误格式相同，包含日期/时间、错误级别以及客户地址、错误信息。但这个CGI错误的错误信息有好几行，这往往会干扰一些错误日志分析软件的工作。

有了这个错误信息，即使是对Perl不太熟悉的人也能够找出许多有关错误的信息，例如至少可以方便地得知是哪几行代码出现了问题。Perl在报告程序错误方面的机制是相当完善的。当然，不同的编程语言输出到错误日志的信息会有所不同。

由于CGI程序运行环境的特殊性，如果没有错误日志的帮助，大多数CGI程序的错误都将很难解决。

有不少人在邮件列表或者新闻组中抱怨说自己有一个CGI程序，当打开网页时服务器却返回错误，比如 “Internal Server Error”。我们可以肯定，这些人还没有看过服务器的错误日志，或者根本不知道错误日志的存在。决多大多数情况下， 错误日志能够精确地指出CGI错误的所在以及如何修正这个错误。

参考：

<http://baike.baidu.com/link?url=kaD1D9eKAdbGQUOPq3aAblmdfWpaVPpf1_G_Tzp3MxdEgwWST94XAehpwjYay4YwoxMrgwc_kPt9htvJfAvqd0erFkd__ln4nGa55pv9ruCidZasZ0vWPzNvO9Lz3Xsw> apache 日志的百度百科

<http://blog.chinaunix.net/uid-21202343-id-1829690.html>

<http://blog.csdn.net/u013699800/article/details/37593491> apache日志详解