Depth 首部，就默认其值是无穷大（就是说，默认会把源目录的整个结构进行复制或移动）。如果 Depth 设置为 0，方法就只作用于资源本身。如果我们是对集合进行复制或移动的话，在目的地就只创建和源具有相同属性的集合——集合内部的成员就不再复制或移动了。

对于 MOVE 方法，Depth 首部的值只允许为无穷大，原因显而易见。

1. Overwrite首部的效果

COPY 和 MOVE 方法也可能使用 Overwrite 首部。Overwrite 首部的值可以是 T 或 F。如果设置为 T 而且目标已存在，就在 COPY 或 MOVE 之前，对目标资源执行 Depth 值为无穷大的 DELETE 操作。如果 Overwrite 标志设置为 F 而目标资源存在，则操作会失败。

442

2. 对属性的COPY/MOVE

当复制集合或元素时，默认会复制其所有属性。不过，请求可以带有可选的 XML 主体来提供额外的操作信息。可以指定要使操作成功，必须成功复制所有属性；或者定义要使操作成功，必须复制哪些属性。

下面有两个特殊状况下的例子。

- 假设把 COPY 或 MOVE 作用到 CGI 程序或者其他产生内容的脚本程序的输出上。为了保持语义，如果由 CGI 脚本产生的文件被复制或移动了，WebDAV 要提供 src 和 link 这两个 XML 元素，指向产生此页面的程序的位置。
- COPY 和 MOVE 方法不一定能够复制所有的活属性。例如，我们来看一个 CGI 程序。如果从 cgi-bin 目录中把它拷贝走，可能就不会再去执行它了。WebDAV 的现有规范让 COPY 和 MOVE 实现的是“尽力而为”解决方案，复制所有的静态属性和合适的活属性。

3. 被锁定的资源与COPY/MOVE

如果资源目前正被锁定，COPY 和 MOVE 都禁止把锁移动或复制到目标上。在这两种情况下，如果要在一个自己有锁的现存集合中创建目标，所复制或移动的资源就会被加到那个锁中。请看下面的例子：

```
COPY /publishing HTTP/1.1
Destination: http://minstar/archived/publishing-old
```

假设 /publishing 和 /archived 分别处于两个不同的锁之下：lock1 和 lock2。当 COPY