**HTTP协议基础**

HTTP（HyperTextTransferProtocol）是超文本传输协议的缩写，它用于传送WWW方式的数据，关于HTTP协议的详细内容请参考RFC2616。HTTP协议采用了请求/响应模型。客户端向服务器发送一个请求，请求头包含请求的方法、URI、协议版本、以及包含请求修饰符、客户信息和内容的类似于MIME的消息结构。服务器以一个状态行作为响应，相应的内容包括消息协议的版本，成功或者错误编码加上包含服务器信息、实体元信息以及可能的实体内容。

通常HTTP消息包括客户机向服务器的请求消息和服务器向客户机的响应消息。这两种类型的消息由一个起始行，一个或者多个头域，一个只是头域结束的空行和可选的消息体组成。HTTP的头域包括通用头，请求头，响应头和实体头四个部分。每个头域由一个域名，冒号（:）和域值三部分组成。域名是大小写无关的，域值前可以添加任何数量的空格符，头域可以被扩展为多行，在每行开始处，使用至少一个空格或制表符。

通用头域

通用头域包含请求和响应消息都支持的头域，通用头域包含Cache-Control、Connection、Date、Pragma、Transfer-Encoding、Upgrade、Via。对通用头域的扩展要求通讯双方都支持此扩展，如果存在不支持的通用头域，一般将会作为实体头域处理。下面简单介绍几个在UPnP消息中使用的通用头域。

Cache-Control头域

Cache-Control指定请求和响应遵循的缓存机制。在请求消息或响应消息中设置Cache-Control并不会修改另一个消息处理过程中的缓存处理过程。请求时的缓存指令包括no-cache、no-store、max-age、max-stale、min-fresh、only-if-cached，响应消息中的指令包括public、private、no-cache、no-store、no-transform、must-revalidate、proxy-revalidate、max-age。各个消息中的指令含义如下：

Public指示响应可被任何缓存区缓存。

Private指示对于单个用户的整个或部分响应消息，不能被共享缓存处理。这允许服务器仅仅描述当用户的部分响应消息，此响应消息对于其他用户的请求无效。

no-cache指示请求或响应消息不能缓存

no-store用于防止重要的信息被无意的发布。在请求消息中发送将使得请求和响应消息都不使用缓存。

max-age指示客户机可以接收生存期不大于指定时间（以秒为单位）的响应。

min-fresh指示客户机可以接收响应时间小于当前时间加上指定时间的响应。

max-stale指示客户机可以接收超出超时期间的响应消息。如果指定max-stale消息的值，那么客户机可以接收超出超时期指定值之内的响应消息。

Date头域

Date头域表示消息发送的时间，时间的描述格式由rfc822定义。例如，Date:Mon,31Dec200104:25:57GMT。Date描述的时间表示世界标准时，换算成本地时间，需要知道用户所在的时区。

Pragma头域

Pragma头域用来包含实现特定的指令，最常用的是Pragma:no-cache。在HTTP/1.1协议中，它的含义和Cache-Control:no-cache相同。

请求消息

请求消息的第一行为下面的格式：

MethodSPRequest-URISPHTTP-VersionCRLFMethod表示对于Request-URI完成的方法，这个字段是大小写敏感的，包括OPTIONS、GET、HEAD、POST、PUT、DELETE、TRACE。方法GET和HEAD应该被所有的通用WEB服务器支持，其他所有方法的实现是可选的。GET方法取回由Request-URI标识的信息。HEAD方法也是取回由Request-URI标识的信息，只是可以在响应时，不返回消息体。POST方法可以请求服务器接收包含在请求中的实体信息，可以用于提交表单，向新闻组、BBS、邮件群组和数据库发送消息。

SP表示空格。Request-URI遵循URI格式，在此字段为星号（\*）时，说明请求并不用于某个特定的资源地址，而是用于服务器本身。HTTP-Version表示支持的HTTP版本，例如为HTTP/1.1。CRLF表示换行回车符。请求头域允许客户端向服务器传递关于请求或者关于客户机的附加信息。请求头域可能包含下列字段Accept、Accept-Charset、Accept-Encoding、Accept-Language、Authorization、From、Host、If-Modified-Since、If-Match、If-None-Match、If-Range、If-Range、If-Unmodified-Since、Max-Forwards、Proxy-Authorization、Range、Referer、User-Agent。对请求头域的扩展要求通讯双方都支持，如果存在不支持的请求头域，一般将会作为实体头域处理。

典型的请求消息：

GEThttp://download.microtool.de:80/somedata.exe

Host:download.microtool.de

Accept:\*/\*

Pragma:no-cache

Cache-Control:no-cache

Referer:http://download.microtool.de/

User-Agent:Mozilla/4.04[en](Win95;I;Nav)

Range:bytes=554554-

上例第一行表示HTTP客户端（可能是浏览器、下载程序）通过GET方法获得指定URL下的文件。棕色的部分表示请求头域的信息，绿色的部分表示通用头部分。

Host头域

Host头域指定请求资源的Intenet主机和端口号，必须表示请求url的原始服务器或网关的位置。HTTP/1.1请求必须包含主机头域，否则系统会以400状态码返回。

Referer头域

Referer头域允许客户端指定请求uri的源资源地址，这可以允许服务器生成回退链表，可用来登陆、优化cache等。他也允许废除的或错误的连接由于维护的目的被追踪。如果请求的uri没有自己的uri地址，Referer不能被发送。如果指定的是部分uri地址，则此地址应该是一个相对地址。

Range头域

Range头域可以请求实体的一个或者多个子范围。例如，

表示头500个字节：bytes=0-499

表示第二个500字节：bytes=500-999

表示最后500个字节：bytes=-500

表示500字节以后的范围：bytes=500-

第一个和最后一个字节：bytes=0-0,-1

同时指定几个范围：bytes=500-600,601-999

但是服务器可以忽略此请求头，如果无条件GET包含Range请求头，响应会以状态码206（PartialContent）返回而不是以200（OK）。

User-Agent头域

User-Agent头域的内容包含发出请求的用户信息。

响应消息

响应消息的第一行为下面的格式：

HTTP-VersionSPStatus-CodeSPReason-PhraseCRLF

HTTP-Version表示支持的HTTP版本，例如为HTTP/1.1。Status-Code是一个三个数字的结果代码。Reason-Phrase给Status-Code提供一个简单的文本描述。Status-Code主要用于机器自动识别，Reason-Phrase主要用于帮助用户理解。Status-Code的第一个数字定义响应的类别，后两个数字没有分类的作用。第一个数字可能取5个不同的值：

1xx:信息响应类，表示接收到请求并且继续处理

2xx:处理成功响应类，表示动作被成功接收、理解和接受

3xx:重定向响应类，为了完成指定的动作，必须接受进一步处理

4xx:客户端错误，客户请求包含语法错误或者是不能正确执行

5xx:服务端错误，服务器不能正确执行一个正确的请求

响应头域允许服务器传递不能放在状态行的附加信息，这些域主要描述服务器的信息和Request-URI进一步的信息。响应头域包含Age、Location、Proxy-Authenticate、Public、Retry-After、Server、Vary、Warning、WWW-Authenticate。对响应头域的扩展要求通讯双方都支持，如果存在不支持的响应头域，一般将会作为实体头域处理。

典型的响应消息：

HTTP/1.0200OK

Date:Mon,31Dec200104:25:57GMT

Server:Apache/1.3.14(Unix)

Content-type:text/html

Last-modified:Tue,17Apr200106:46:28GMT

Etag:"a030f020ac7c01:1e9f"

Content-length:39725426

Content-range:bytes554554-40279979/40279980

上例第一行表示HTTP服务端响应一个GET方法。棕色的部分表示响应头域的信息，绿色的部分表示通用头部分，红色的部分表示实体头域的信息。

Location响应头

Location响应头用于重定向接收者到一个新URI地址。

Server响应头

Server响应头包含处理请求的原始服务器的软件信息。此域能包含多个[产品](http://tech.163.com/production/)标识和注释，产品标识一般按照重要性排序。

实体

请求消息和响应消息都可以包含实体信息，实体信息一般由实体头域和实体组成。实体头域包含关于实体的原信息，实体头包括Allow、Content-Base、Content-Encoding、Content-Language、Content-Length、Content-Location、Content-MD5、Content-Range、Content-Type、Etag、Expires、Last-Modified、extension-header。extension-header允许客户端定义新的实体头，但是这些域可能无法未接受方识别。实体可以是一个经过编码的字节流，它的编码方式由Content-Encoding或Content-Type定义，它的长度由Content-Length或Content-Range定义。

Content-Type实体头

Content-Type实体头用于向接收方指示实体的介质类型，指定HEAD方法送到接收方的实体介质类型，或GET方法发送的请求介质类型Content-Range实体头

Content-Range实体头用于指定整个实体中的一部分的插入位置，他也指示了整个实体的长度。在服务器向客户返回一个部分响应，它必须描述响应覆盖的范围和整个实体长度。一般格式：

Content-Range:bytes-unitSPfirst-byte-pos-last-byte-pos/entity-legth

例如，传送头500个字节次字段的形式：Content-Range:bytes0-499/1234如果一个http消息包含此节（例如，对范围请求的响应或对一系列范围的重叠请求），Content-Range表示传送的范围，Content-Length表示实际传送的字节数。

Last-modified实体头

Last-modified实体头指定服务器上保存内容的最后修订时间。