

文章翻译免责声明

本UPnP文档最初用英文发布，并仅有英文版本通过了UPnP论坛的正式审核。文章已采用翻译服务和翻译技术在英文版本的基础上翻译成目标语言版本，UPnP Implementers Corporation及其相关机构不对翻译版本做出任何保证，也不为由翻译不准确所导致的直接或间接损失承担责任。在使用翻译版本中所包括的技术信息时，用户同意UPnP Implementers Corporation和UPnP论坛成员对于英文到目标语言翻译的不完整、或不准确导致的全部或部分损失不承担任何责任。此外，用户同意应保护UPnP Implementers Corporation免受由语言翻译而带来的伤害。

1.0 版，2000 年 6 月 8 日上午 10:41

版权所有© 1999-2000 UPnP™论坛贡献成员。所有权利受到保护。

目录

介绍

0. 寻址

1. 发现

2. 描述

3. 控制

4. 事件触发

5. 展示

术语表

介绍

何为 UPnP?

UPnP 是针对智能家电、无线设备以及各种外观尺寸的个人电脑的普遍对等 (peer-to-peer) 网络连接而设计的一种架构。它旨在为家庭、小型企业、公共场所中或连接到互联网的 ad-hoc 网或未管理网络提供易于使用、灵活且基于标准的连接。UPnP 是一个充分利用 TCP/IP 和 Web 技术的分布式开放型网络体系结构，除能够在家中、办公室和公共场所联网设备之间的完整控制和数据传输之外，还可建立无缝紧密的连接网络。

UPnP 不仅仅只是即插即用外设模式的简单扩展。它设计用于支持零配置、“不可见”联网，以及对众多厂商的广泛设备类型的自动发现。这就意味着，一台设备能够动态加入一个网络，获取一个 IP 地址，通报其功能，以及了解其它设备的存在和功能。DHCP 和 DNS 服务器为可选服务器，仅当他们在网上存在时可以使用。最后，设备能够顺利地自动离线，而不会造成任何不期望的影响。

UPnP 充分利用了包括 IP、TCP、UDP、HTTP 和 XML 在内的互联网组件。正如互联网那样，合约基于陈述性的有线协议，以 XML 来表达，并通过 HTTP 进行传输。IP 网间协议凭借其以下已经被验证的能力而成为 UPnP 的一个有力选择：跨越不同的物理媒体、支持实际的多厂商互操作，以及实现与互联网、大量家庭和办公室内联网的协作融合等等。UPnP 的设计明确用于支持这些环境。此外，当成本、技术或传统因素等阻止与 UPnP 连接的媒体或设备运行 IP 协议时，UPnP 还可通过桥接方式支持运行非 IP 协议的媒体。

何为 UPnP 的“通用性”？不使用设备驱动程序；取而代之的是通用协议。UPnP 网络不依赖于任意媒体。UPnP 设备可以在任何操作系统上采用任何编程语言来实现。UPnP 并未针对运行于控制点上的应用而指明或限制 API 的设计；操作系统厂商可以创建满足其客户需求的 API。UPnP 通过使用浏览器和传统应用程序控制来使厂商能够控制设备的用户界面（UI）并实现交互。

UPnP 论坛

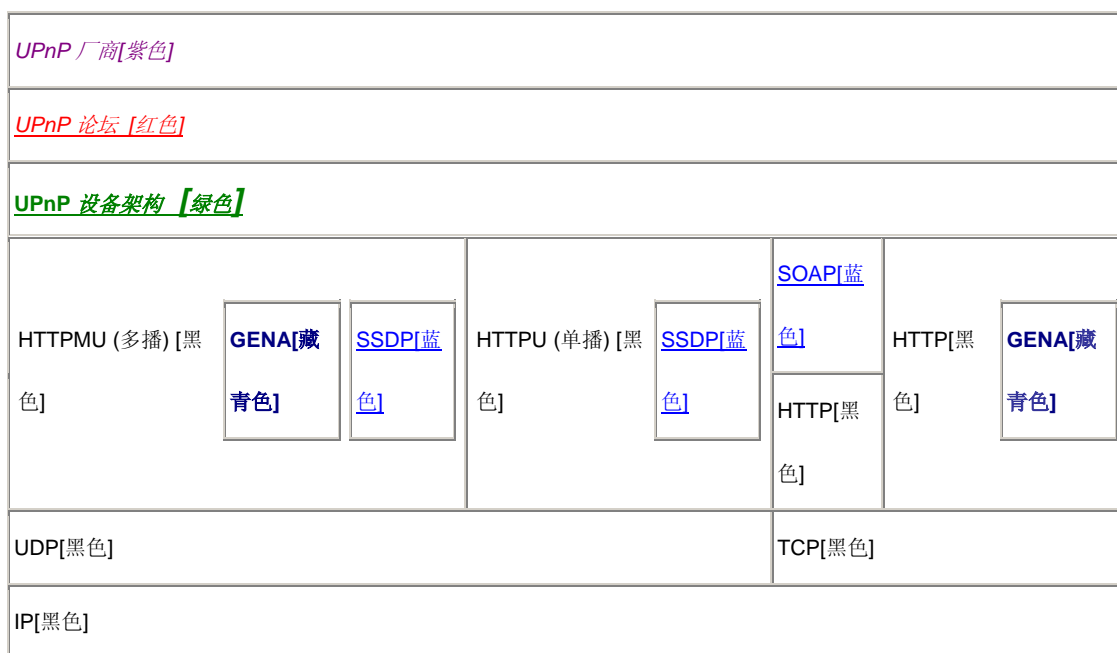
UPnP 论坛是一项业界发起的计划，致力于在众多不同厂商的独立设备和个人计算机之间轻松建立强健的连接。UPnP 论坛寻求开发描述设备协议和基于 XML 的设备模式的标准，以便在一个可伸缩的网络环境中实现设备间互操作性。UPnP 论坛负责监督针对一致性设备的一项标识计划。

UPnP 论坛设立了具体专业知识方面的工作委员会。这些工作委员会负责制定建议设备标准、构建范例实施，以及构建适当的测试套件。本文指出了属于 UPnP 论坛工作委员会范围的具体技术决定。

UPnP 厂商能够凭借共享知识产权和标识计划的优势和互操作性的信心构建出符合一致性要求的设备。如不参加标识计划，厂商还可以构建符合 UPnP 设备架构的设备，而不用通过正式标准程序。如果厂商构建非标准设备，将由他们自己制定技术决策，而非 UPnP 论坛工作委员会。

本文内容

此处包含的 UPnP 设备架构（原称作 DCP 框架）定义了控制器或控制点以及设备之间的通信协议。UPnP 针对发现、描述、控制、事件触发和展示采用了以下协议栈。



在最高一层，消息在逻辑上仅仅包含关于厂商设备的 UPnP 厂商特定信息。移至下一层协议栈后，厂商内容由 UPnP 论坛工作委员会定义之信息提供。来自以上各层的消息存储在 UPnP 特定协议中，并在本文中进行了定义。之后，

以上消息通过采用简单服务发现协议（SSDP）、通用事件通知架构（GENA）和简单对象访问协议（SOAP）来进行格式化。然后消息通过运行于 UDP 上的多播或单播类 HTTP，或是运行于 TCP 上的标准 HTTP 进行传输。最终，以上所有消息均通过 IP 进行传输。本文其余部分详细描述了这些协议层中每层的内容和格式。为便于参考，以上[方括弧]中的颜色指出了全文各协议所定义的具体消息成份。

UPnP 网络互连的基础是 IP 寻址。每台设备均必须配有动态主机配置协议（DHCP）客户端，并在设备首次与网络连接时搜索 DHCP 服务器。如果 DHCP 服务器可以使用，即网络处于管理状态，则设备必须采用分配给它的 IP 地址。如果没有 DHCP 服务器可用，即网络处于未管理状态，则设备必须利用 Auto IP 来获取一个地址。简言之，Auto IP 说明了一台设备如何从一组保留地址中智能地选出一个 IP 地址，以及如何能够在处于管理和未管理状态的网络间轻松移动。如果设备在 DHCP 交易过程中获得了一个域名（例如通过一台 DNS 服务器或通过 DNS 转发），则设备应当在后来的网络操作中采用该名称；否则即应采用其 IP 地址。

如果获取了一个 IP 地址，则 UPnP 网络的第 1 步是发现。在将一个设备添加到网络上之后，UPnP 发现协议允许该设备向网络中的控制点宣告其服务。同样，当一个控制点被添加到网络后，UPnP 发现协议允许该控制点在网上搜索感兴趣的设备。两种情况下的根本信息交换均为一个发现消息，包含有关该设备或其服务之一的一些基础信息（例如其类型、标识符和指向更详细信息的一个指针）。UPnP 发现协议基于简单服务发现协议（SSDP）。以下发现部分说明了设备如何进行宣告，控制点如何搜索，以及有关发现消息格式的详细信息。

UPnP 网络中的第 2 步是描述。控制点在发现一个设备之后仍然对其知之甚少。为了使控制点了解到更多关于设备及其能力的信息或与设备进行交互，则控制点必须取得来自该设备在发现消息中所提供之 URL 的设备描述。设备可能包含其它逻辑设备，以及功能单元或服务。对于设备的 UPnP 描述通过 XML 来表

达，并包括诸如模型名称和号码、序列号、制造商名称和厂商专门网站 **URL** 等专门针对厂商的制造商信息。该描述还包括一系列任意的嵌入式设备或服务，以及用于控制、事件触发和展示的 **URL**。对于每项服务，此描述均包括一系列命令或动作，而服务（参数或变量）对于每个动作做出响应；针对服务的描述还包括一系列变量；这些变量模型化服务在运行时的状态，并通过数据类型、范围和事件特征进行描述。以下关于描述的部分说明了设备如何被描述，以及这些描述如何被控制点取得。

UPnP 网络中的第 3 步是控制。当一个控制点取得设备描述后，该控制点可将动作发至一个设备的服务。为此，控制点将一条适当的控制消息发至服务的控制 **URL**（在设备描述中提供）。控制消息同样利用简单对象访问协议（**SOAP**）通过 **XML** 来表达。类似于功能调用，该服务针对控制消息返回了所有的专门动作取值。动作的效果可以通过描述服务运行时状态的变量进行描述。以下关于控制的部分说明了有关动作、状态变量以及控制消息格式的描述。

UPnP 网络的第 4 步是事件触发。针对服务的 **UPnP** 描述包括一个服务响应的动作列表，以及一个对服务器运行时状态进行展示的变量列表。在这些变量变更时服务会发布更新，一个控制点可以预订接收此信息。服务通过发送事件消息来发布更新。事件消息包含一个或多个状态变量名和这些变量的当前值。这些消息同样通过 **XML** 来表达，并采用通用事件通知架构（**GENA**）格式。当控制点首次预定时，会发送一个特殊的初始事件消息；此事件消息包含所有事件变量的名称和值，并允许订阅者对服务状态模式进行初始化。为了支持拥有多个控制点的环境，事件触发设计用于将任何动作的效果通知所有控制点。因此，所有订阅者均会收到全部的事件消息。订阅者收到关于所有已变更事件变量的事件消息，此事件消息无论状态变量为何改变都被发送（由于响应一个要求动作，或由于服务建模状态的变更）。以下关于事件触发的部分说明了事件消息的预订和格式。

UPnP 网络中的第 5 步是展示。如果设备有用于展示的 **URL**，那么控制点就可以通过此 **URL** 取得一个页面，在浏览器中加载该页面，并且根据页面的功

能，支持用户控制设备和/或浏览设备状态。每一项完成的程度取决于展示页面和设备的具体功能。以下关于展示的部分说明了关于取得一个展示页面的协议。

读者对象

本文的读者对象包括 UPnP 设备厂商、UPnP 论坛工作委会的成员，以及需要了解 UPnP 协议技术细节的其他人员。

本文假定读者熟悉 HTTP、TCP、UDP 和 IP 协议族；本文无意对这些协议进行说明。本文还假定多数读者并不了解 XML。虽然它并非一份 XML 指南，但考虑到 XML 与 UPnP 的紧密关系，文章仍详细介绍了 XML 相关问题。本文没有假定读者了解各种编程或脚本语言。

要求 Vs 推荐使用

本文中，特性被描述为要求、推荐使用或可选，表示如下：

要求（或必须）

这些基本特性必须依照 UPnP 来实现。

推荐使用（或应当）

这些特性添加了 UPnP 支持的功能，应当进行实现。推荐使用的特性充分利用了 UPnP 的能力，通常不会造成成本的大幅提高。注意，对于一致性测试而言，如果要实施一个推荐特性，则该特性必须满足指定要求，以符合这些准则。一些推荐使用的特性在将来可能变成要求特性。

可选（或可能）

这些特性既非 UPnP 要求，也非其推荐使用，然而一旦实现，则其必须满足指定要求，以符合这些准则。这些特性在将来不可能成为要求特性。

术语缩写

术语缩写	含意	术语缩写	含意
ARP	地址解析协议	SSDP	简单服务发现协议
DHCP	动态主机配置协议	UPC	通用产品代码
DNS	域名系统	UPnP	UPnP
FXPP	灵活的 XML 处理框架	URI	统一资源标识符
GENA	通用事件通知架构	URL	统一资源定位器
HTML	超文本标记语言	URN	统一资源名称
HTTPMU	基于 UDP 的 HTTP 多播	UUID	全球唯一标识符
HTTPU	基于 UDP 的 HTTP（单播）	XML	扩展标记语言
ICANN	指定名称和编号的互联网公司		
SOAP	简单对象访问协议（SOAP）		

参考和资源

RFC2616

HTTP：超文本传输协议 1.1 IETF RFC。

<http://search.ietf.org/rfc/rfc2616.txt?number=2616>。

RFC2279

UTF-8，ISO 10646 的一种转化形式（字符编码）。IETF RFC。

<http://search.ietf.org/rfc/rfc2279.txt?number=2279>

XML

扩展标记语言，W3C 推荐。 <http://www.w3.org/XML/>。

本文每部分均包含关于特定主题资源的额外信息。

致谢

微软 UPnP 团队对于代表 UPnP 论坛指导委员会和 UPnP 厂商团体的众多技术评论家所给予我们的鼎力相助深表感谢。这些评论家凭借其丰富的技术信息、远见卓识和聪明才智为撰写本文提供了大力支持。

我们同样感谢微软的软件工程师、测试员和计划经理，他们提供了宝贵的反馈和技术内容，以确保本文信息的准确性和及时性。

0. 寻址

寻址是 UPnP 网络的第 0 步。设备通过寻址来获得一个网络地址。寻址支持发现（第 1 步，控制点发现感兴趣的设备）、描述（第 2 步，控制点对设备功能进行了解）、控制（第 3 步，一个控制点将命令发至设备）、事件触发（第 4 步，控制点监听设备状态的变化），以及展示（第 5 步，控制点显示设备的一个用户界面）。

UPnP 网络的基础在于 IP 寻址。每个设备均必须配有一个动态主机配置协议（DHCP）客户端，并在此设备首次与网络相连时搜索一台 DHCP 服务器。如果有一个可用的 DHCP 服务器，即网络处于管理状态，则此设备必须采用分配给它的 IP 地址。如果没有可用的 DHCP 服务器，即网络处于未管理状态，则此设备必须利用自动 IP 寻址（Auto-IP）来获取一个地址。

Auto-IP 定义了设备如何：**(a)** 在 DHCP 不可用时做出决定，以及 **(b)** 从一组本地链接的 IP 地址中智能地选出一个 IP 地址。这种地址分配方法使设备能够在处于管理和未管理状态的网络之间轻松移动。

此部分所描述的操作在下列参考文件中得到了进一步阐明。如果本文和参考文件存在冲突之处，则始终以参考文件为准。

0.1 寻址：决定是否采用 Auto-IP

支持 AUTO-IP 并针对动态地址分配而配置的一个设备，通过发出一条 DHCPDISCOVER 消息向 DHCP 请求 IP 地址开始运行。此 DHCP 客户端所应监听 DHCPOFFERS 的时间总长取决于具体实现。如果在此期间收到一条 DHCPOFFER，则此设备必须继续进行动态地址分配。如果没有收到有效的 DHCPOFFERS，则此设备可以自动配置一个 IP 地址。

0.2 寻址：选择一个地址

要利用 Auto-IP 来自动配置一个 IP 地址，设备采用了一个取决于具体情况的算法，以便在 169.254/16 范围内选出一个地址。此范围内的前后 256 个地址需要保留，不得使用。

接下来必须对选定地址进行测试，以确定该地址是否已被使用。如果该地址已被其它设备使用，则必须选出和测试另一地址，重试次数取决于具体的实现。地址的选择应当随机进行，以免在多个设备试图分配地址时产生冲突。

0.3 寻址：测试地址

为了测试选出的地址，设备必须采用地址解析协议（ARP）来进行探测。地址解析协议（ARP）探测是一个 ARP 请求，其中设备硬件地址用作发送者的硬件地址，而发送者的 IP 地址设为多个 0。设备继而会监听对于该 ARP 探测，或同一 IP 地址的其它 ARP 探测所做出的响应。如果发现任意一个 ARP 数据包，则此设备必须考虑正在使用的地址，并尝试一个新地址。

0.4 寻址：定期检查动态地址的可用性

自动配置了 IP 地址的设备必须进行定期检查，以确定一台 DHCP 服务器的存在。这一过程通过发送 DHCPDISCOVER 消息来完成。该项检查的频度取决于具体实现，然而每 5 分钟检查一次将可维持要求网络带宽和连接性维护之间的平衡。如果收到 DHCP 信息，设备必须继续进行动态地址分配。一旦 DHCP 指定地址到位，则设备即可释放该自动配置的地址，或者也可选择将该地址保留一段时间以维持连接性。

要从一个 IP 地址转到一个新地址，设备必须取消所有已经公开的宣告并重新发布新宣告。关于发现的部分说明了宣告及其取消。

0.5 寻址：设备命名与 DNS 交互

设备一旦具有了一个有效的网络 IP 地址，即可通过该地址在该网络中被定位和访问。可能在某些情况下用户需要寻找和识别一个设备。在这些情况下，对使用者而言，设备的一个友好的名称要比一个 IP 地址易于使用得多。

此外，名称远比 IP 地址要稳定。通过名称查询设备的客户在设备的 IP 地址变更时无需做出任何修改。设备的 DNS 名称对其 IP 地址的映射可以按照 RFC2136 被手动或动态输入 DNS 数据库。支持动态 DNS 更新的计算机和设备能够将其 DNS 记录直接记入 DNS 中。同时还有可能配置一台 DHCP 服务器来代表这些 DHCP 客户端登记 DNS 记录。

0.6 寻址：名称到 IP 地址的解析

当一台计算机需要访问另一台以 DNS 名称来指明的设备时，它要发现其 IP 地址。计算机按照 RFC1034 和 1035 将一个 DNS 询问提交至预先配置的 DNS 服务器，并从包含目标设备 IP 地址的 DNS 服务器收到一个响应。计算机能够静

态地采用 DNS 服务器列表进行预先配置。或者，计算机还能通过 DHCP 采用 DNS 服务器列表进行配置，或在地址分配之后通过一条 DHCPINFORM 消息进行配置。

0.7 寻址参考

Auto-IP

在一个Ad-Hoc IPv4 网络中自动选出一个IP地址。IETF草案。

<http://search.ietf.org/internet-drafts/draft-ietf-dhc-ipv4-autoconfig-05.txt>.

RFC1034

域名——概念与工具。IETF RFC。

<http://search.ietf.org/rfc/rfc1034.txt?number=1034>.

RFC1035

域名——实施与规范。IETF RFC。

<http://search.ietf.org/rfc/rfc1035.txt?number=1035>.

RFC2131

动态主机配置协议IETF RFC。

<http://search.ietf.org/rfc/rfc2131.txt?number=2131>.

RFC2136

域名系统中的动态更新。IETF RFC。

<http://search.ietf.org/rfc/rfc2136.txt?number=2136>.

由 DHCP 客户端和服务端进行的动态 DNS 更新

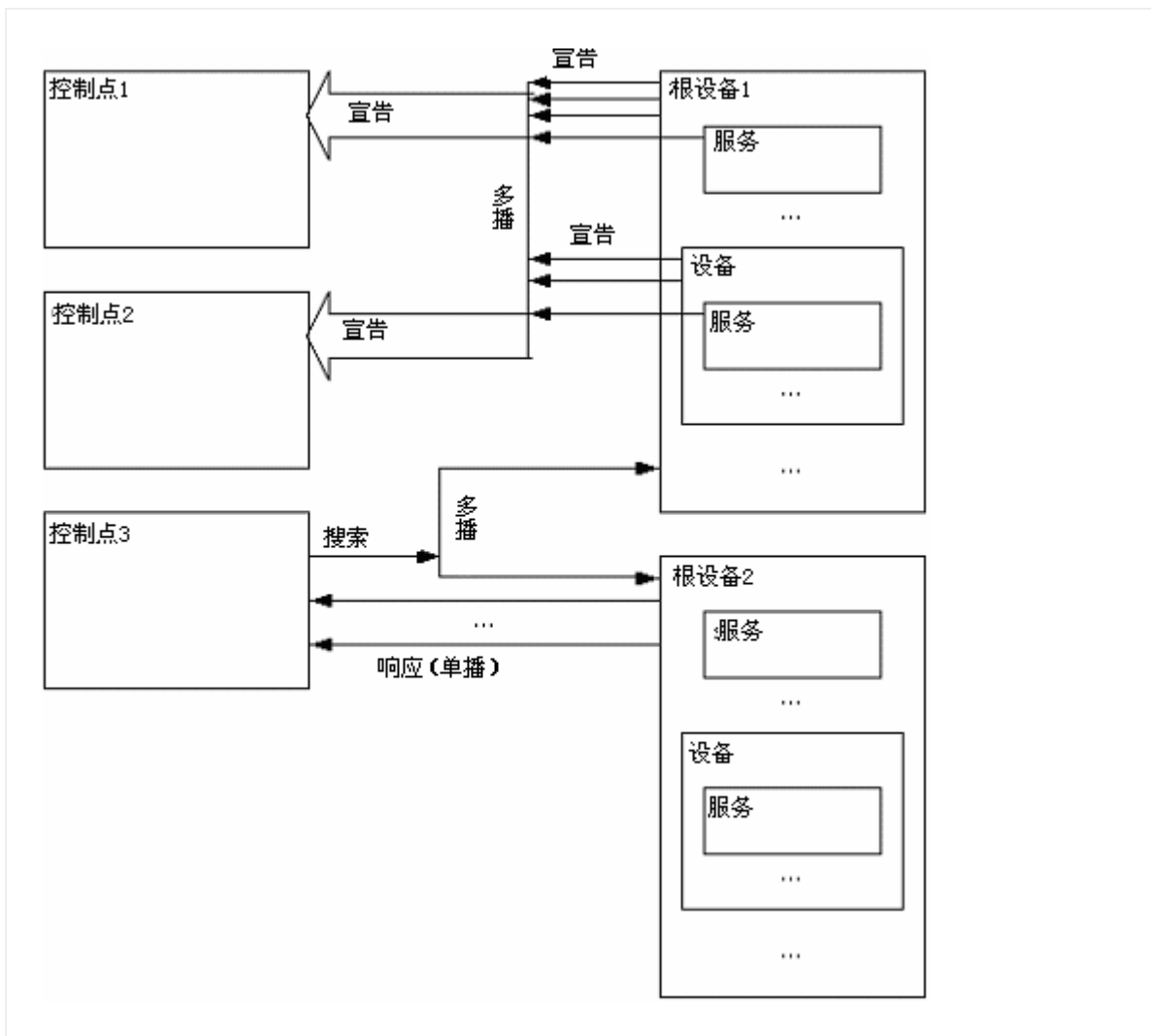
DHCP和DNS间的交互。IETF草案。

<http://search.ietf.org/internet-drafts/draft-ietf-dhc-dhcp-dns-12.txt>.

1. 发现

发现是 UPnP 网络中的第 1 步。发现紧随设备获取一个网络地址的寻址（第 0 步）之后。控制点通过发现来找到感兴趣的设备。发现支持描述（第 2 步，控制点对设备能力进行了解）、控制（第 3 步，一个控制点将命令发至设备）、事件触发（第 4 步，控制点监听设备状态的变化），以及展示（第 5 步，控制点展示设备的一个用户界面）。

发现是 UPnP 网络中的第 1 步。当设备被添加到网络后，UPnP 发现协议允许该设备向网络中的控制点宣告其服务。同样，当一个控制点被添加到网络后，UPnP 发现协议允许该控制点在网搜索感兴趣的设备。两种情况下的基本交换信息均为一条发现消息，其中包含关于该设备或其服务之一的少量基础性的细节信息（例如其类型、标识符和指向更详细信息的一个指针）。



当一个新设备被添加到网络后，它会多播大量的发现消息来宣告其嵌入式设备和服务。所有感兴趣的控制点均能监听标准的多播地址，以获取宣布新能力可用的通知。

同样，当一个新控制点被添加到网络后，它会多播一条发现消息来搜索感兴趣的设备、服务，或同时搜索二者。所有设备均必须监听这些消息的标准多播地址，同时必须在其任何嵌入式设备或服务与发现消息中的搜索标准相符合时做出响应。

需要重申的是，控制点发现感兴趣的设备可能是由于该设备发送了宣告自己的发现消息，或由于该设备对一条搜索设备的发现消息做出了响应。在任何一种情况下，如果控制点对一个设备感兴趣并想了解关于该设备的更多信息，则该控制点必须采用发现消息中的信息来发送一条描述询问消息。关于描述的部分对描述消息做了详细说明。

当设备离线后，它应当多播大量的发现消息来撤消其早期宣告，有效地宣布其嵌入式设备和服务将不再可用。

为了限制网络拥塞，每个 IP 数据包针对每条多播消息的有效期限（TTL）必须默认为 4，且应可以配置。

发现在帮助实现使用不同版本 UPnP 网络的设备与控制点的互操作性方面有着至关重要的作用。UPnP 设备架构（在此定义）用主、副两个版本来描述，通常写作 *major.minor*，且两者均为整数。副版本改进的部分必须兼容属于同一主版本的前副版本。主版本改进的部分不要求是前版的扩展集，并且不保证向后兼容。版本信息在发现和描述消息中进行传送。在前者中，每条发现消息均包括设备支持的 UPnP 网络版本。作为备份，后者同样包括相同的信息。此部分说

明了发现消息中版本信息的格式和针对发现消息的具体要求，以维持与副版本改进部分的兼容性。

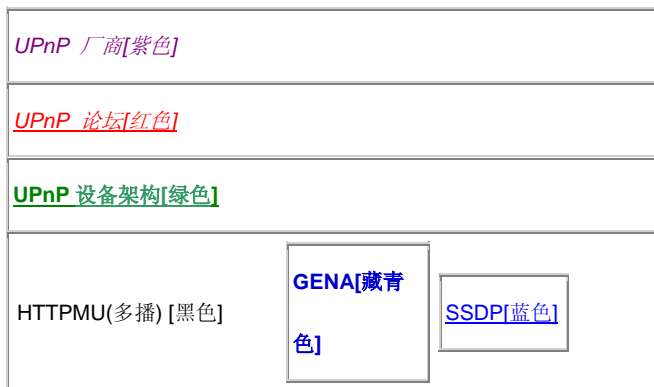
标准的多播地址，以及宣告、搜索和撤消机制由简单服务发现协议（SSDP）来进行定义。此节其余部分详细说明了简单服务发现协议（SSDP），并列举了设备如何宣告和撤消其宣告，以及控制点如何搜索和设备如何响应等等。

1.1 发现：宣告

当设备添加到网络后，UPnP 发现协议允许该设备宣告其服务给控制点。这一点通过将发现消息多播到一个标准地址和端口而得以实现。控制点监听此端口来探测新功能何时在网上可用。为了宣告其全部能力范围，一个设备对应其每个嵌入式设备和服务而需要多播一系列的发现消息。每条消息均包含关于嵌入式设备（或服务）的专门信息，以及关于其内含设备的信息。消息应包括直到宣告过期的持续时间；如果设备仍然可用，则应将宣告一并重新发送（带有新的持续时间）。如果设备不再可用，则设备应当明确撤消其宣告，然而如果设备本身不能撤消，则宣告将自动过期。

1.1.1 发现：宣告协议与标准

为了发送（和接收）宣告消息，设备（与控制点）采用了以下总体 UPnP 协议栈的子集。（总体 UPnP 协议栈在本文开始处列出。）



HTTPMU(多播)[黑色]
UDP[黑色]
IP[黑色]

在最高层，发现消息包含特定厂商信息，例如用于设备描述的 URL 和设备标识符。在协议栈接下来的一层，通过来自 UPnP 论坛工作委员会的信息（如设备类型）对厂商内容进行了补充。来自以上各层的消息存储在 UPnP 特定协议中，并在本文中进行了定义。接下来，以上消息通过一个 HTTP 的多播变量进行传送，该变量已通过采用通用事件通知架构（GENA）方法和标头，以及简单服务发现协议（SSDP）标头而得以扩展。HTTP 消息通过 IP 经 UDP 进行传送。为便于参考，以上[方括弧]中的颜色显示了何种协议定义下面发现消息中的具体标头与值。

1.1.2 发现：宣告：设备可用——ssdp:alive NOTIFY

在设备被加入网络时，它采用多播方式发送发现消息，以告知设备包含的根设备信息，所有嵌入式设备以及它包含的服务。每个发现消息都包含四个主要组成部分：

1. 在 NT 标头中发送的潜在搜索目标（例如设备类型）；
2. 在 USN 标头中发送的宣告复合标识符；
3. 在 LOCATION 标头中发送的关于设备更多信息的 URL 地址（或有关服务的内含设备），以及
4. 在 CACHE-CONTROL 标头中发送的宣告合法持续时间。

为了告知其能力，设备选择多播多个发现消息。特别是，根设备必须多播：

- 对于根设备，存在三种发现消息。

	NT	USN*
1	根设备 UUID**	根设备的 UUID
2	设备类型: 设备版本	根设备的 UUID 和: : 和设备类型: 设备版本
3	UPnP:rootdevice	根设备的 UUID 和: : 和 UPnP:rootdevice

- 对于嵌入式设备，存在两种发现消息。

	NT	USN*
1	嵌入式设备 UUID**	嵌入式设备的 UUID
2	设备类型: 设备版本	嵌入式设备的 UUID 和: : 和设备类型: 设备版本

- 每种服务响应一次。

	NT	USN*
1	服务类型: 服务版本	内含设备的 UUID 和: : 和服务类型: 服务版本

* 注意 USN 标头前缀（位于双冒号前）必须与设备描述中的 UDN 元素值相匹配。（描述部分主要说明 UDN 元素。）

** 注意 UT 标头值必须与设备描述中的 UDN 元素值相匹配。

如果一个根设备有 d 个嵌入式设备， s 个嵌入服务，但是只包含 k 个不同的服务类型，这将会发出 $3+2d+k$ 次请求。这些宣告向感兴趣的控制点描述了设备的所有能力。这些消息必须作为一个系列一起发出，具有大约相接近的过期时间，发送的顺序无关紧要，但是不能对单个消息进行刷新或取消操作。

选择一个适当的持续期是在最小化网络流量和最大化设备状态及时更新之间求得一个平衡。相对较短的持续时间（最短 1800 秒）可以保证控制点在牺牲额外网络流量的情况下获得设备的当前状态；持续期较长（按照时间顺序）会损失设备状态的刷新速度，但是可以大大减少网络流量。一般而言，设备厂商应该选择一个与预期设备使用情况相对应的值：希望短期内成为网络组成部分的设备持续时间短；希望成为网络长期成员的设备持续时间长。

鉴于 UDP 的不可靠特性，设备应该多次发送上述设备发现消息。而且为了排除控制点无法收到设备或服务宣告的可能性，设备应该定期重新发送它的宣告（参见：下面的 **CACHE-CONTROL**）。注意：UDP 信息包的长度有一定的限制（在有些实现方案中可能只有 512 字节），并且不保证以上 $3+2d+k$ 条消息以特殊顺序到达。

在设备加入网络时，它必须用 **NOTIFY** 方法发送一个多播请求，并且 **NTS** 标头中的 **ssdp:alive** 采用以下格式。用 *斜体* 表示的值为实际值的占位符。

```
NOTIFY * HTTP/1.1  
HOST: 239.255.255.250:1900  
CACHE-CONTROL: max-age = seconds until advertisement expires  
LOCATION: URL for UPnP description for root device  
NT: search target  
NTS: ssdp:alive  
SERVER: OS/version UPnP/1.0 product/version  
USN: advertisement UUID
```

（**NOTIFY** 方法发送的请求没有消息体，但请注意，消息与最后一个 **HTTP** 标头之间必须空一行。）

IP 信息包的 **TTL** 必须缺省为 **4**，并且应该可以配置。

下面的列表提供了上面出现的命令行与标头的详细信息。除特别注明外，所有标头值均要区分大小写。

命令行

NOTIFY

GENA 定义的用以发送通知和事件的方法。

*

请求通用，不受具体资源的限制。必须是*。

HTTP/1.1

HTTP 版本。

标头

HOST

要求。由互联网编号分配组织（IANA）为 SSDP 保留的多播信道和端口。

必须是 [239.255.255.250:1900](#)。

CACHE-CONTROL

要求。必须带有 **max-age**，指代宣告有效存活时间。如果超过此时间间隔，控制点可以认为设备（或服务）不再可用。应大于 1800 秒（30 分钟）。

具体由 UPnP 厂商决定。取整数。

LOCATION

要求。包含根设备 UPnP 描述的 URL 地址。在某些未受管理的网络中，URL 主机可能包含 IP 地址（与域名相对）。具体由 UPnP 厂商决定。单一 URL。

NT

GENA 规定使用的标头。通知类型。必须采用以下一种形式。（参见：上表。）单一 URI。

UPnP:rootdevice

向根设备发送一次。

uuid:*device-UUID*

向每种设备（根设备或嵌入式设备）发送一次。设备 UUID 由 UPnP 厂商指定。

urn: schemas-UPnP-org:device:*deviceType:v*

向每种设备（根设备或嵌入式设备）发送一次。设备类型与版本由 UPnP 论坛工作委员会定义。

urn: schemas-UPnP-org:service:*serviceType:v*

向每种服务发送一次。服务类型与版本由 UPnP 论坛工作委员会定义。

NTS

GENA 规定使用的标头。通知子类型。必须是 [ssdp:alive](#)。单一 URI。

SERVER

要求。包含操作系统名称、操作系统版本、[UPnP/1.0](#)、产品名称以及产品版本等信息。具体由 UPnP 厂商决定。字符串。

USN

SSDP 要求使用的标头。唯一服务名称。必须是以下一种。（参见：上表。）前缀（位于双冒号前）必须与设备描述中的 UDN 元素值相匹配。（描述部分主要说明 UDN 元素。）单一 URI。

uuid:*device-UUID*::[UPnP:rootdevice](#)

向根设备发送一次。设备 UUID 由 UPnP 厂商指定。

uuid:*device-UUID*

向每种设备（根设备或嵌入式设备）发送一次。设备 UUID 由 UPnP 厂商指定。

uuid:*device-UUID*::urn:[schemas-UPnP-org:device:deviceType:v](#)

向每种设备（根设备或嵌入式设备）发送一次。设备 UUID 由 UPnP 厂商指定。设备类型与版本由 UPnP 论坛工作委员会定义。

uuid:*device-UUID*::urn:[schemas-UPnP-org:service:serviceType:v](#)

向每种服务发送一次。设备 UUID 由 UPnP 厂商指定。服务类型与版本由 UPnP 论坛工作委员会定义。

（NOTIFY 方法发送的请求没有响应消息。）

1.1.3 发现：宣告：设备不可用——ssdp:byebye NOTIFY

在设备及其服务将要从网络中退出时，设备应该对于每个未超期的 [ssdp:alive](#) 消息以多播方式传送 [ssdp:byebye](#) 消息。但如果设备突然从网络退出，

它可能来不及发出这个通知消息。因此，发现消息必须在 **CACHE-CONTROL** 标头中包含超时值（如上所述）；如果不重新发出宣告，发现消息最后会因超时而必须从控制点的高速缓存中除去。

（注：当控制点将要从网络中退出时，不需要采取与发现相关的动作。）

当一个设备即将从网络中退出时，它应该为每一条 **ssdp:alive** 消息发送一条多播请求，以明确撤销其发现消息。每个多播请求必须采用 **NOTIFY** 方法，并且 **NTS** 标头中的 **ssdp:byebye** 采用以下格式。用斜体表示的值为实际值的占位符。

```
NOTIFY * HTTP/1.1  
HOST: 239.255.255.250:1900  
NT:search target  
NTS:ssdp:byebye  
USN:advertisement UUID
```

（**NOTIFY** 方法发送的请求没有消息体，但请注意，消息与最后一个 **HTTP** 标头之间必须空一行。）

IP 信息包的 **TTL** 必须缺省为 **4**，并且可以配置。

下面的列表提供了上面出现的命令行与标头的详细信息。除特别注明外，所有标头值均要区分大小写。

命令行

NOTIFY

GENA 定义的用以发送通知和事件的方法。

*

请求通用，不受具体资源的限制。必须是*。

HTTP/1.1

HTTP 版本。

标头

HOST

要求。由互联网编号分配组织 (IANA) 为 SSDP 保留的多播信道和端口。必须是 239.255.255.250:1900。

NT

GENA 规定使用的标头。通知类型。（参见以上 `ssdp:alive NOTIFY` 中 NT 标头的要求值列表。）单一 URI。

NTS

GENA 规定使用的标头。通知子类型。必须是 ssdp:byebye。单一 URI。

USN

SSDP 要求使用的标头。唯一服务名称。（参见以上 `ssdp:alive NOTIFY` 中 USN 标头的要求值列表。）单一 URI。

（NOTIFY 方法发送的请求没有响应消息。）

鉴于 UDP 的不可靠性，设备应该发送多次以上消息。因此，如果控制点无法获得设备或服务不可用的通知，原始发现消息最终将会因过期失效而被移除。

1.2 发现：搜索

当一个控制点加入到网络中时，UPnP 发现协议允许控制点寻找网络上感兴趣的设备。消息搜索是通过多播方式实现的，包括类型、或目标，相当于设备或服务的类型或标识符。设备响应消息包括的发现消息本质上与新连接设备的宣告消息相同，前者是单播，而后者是多播。

1.2.1 发现：搜索协议与标准

为了搜索设备（并被控制点发现），控制点（和设备）使用以下总体 UPnP 协议栈的子集。（总体 UPnP 协议栈在本文开始处列出。）

UPnP 厂商[紫色]	
UPnP 论坛[红色]	
UPnP 设备架构[绿色]	
HTTPU（单播）（黑色） SSDP[蓝色]	HTTPMU（多播）（黑色） SSDP[蓝色]
UDP（黑色）	
IP[黑色]	

在最高层，搜索消息包含特定厂商信息，如控制点、设备和服务标识符。在协议栈接下来的一层，通过来自 UPnP 论坛工作委员会的信息（如设备或服务类型）对厂商内容进行了补充。来自以上各层的消息存储在 UPnP 特定协议中，并在本文中进行了定义。按照先后顺序，通过 HTTP 多播变量分别传送搜索请求，该变量已经通过简单服务发现协议（SSDP）方法标头得到了扩展。搜索响应通过 HTTP 单点变量（已经采用 SSDP 进行了扩展）完成传送。（当控制点搜索设备时无需 GENA。）两种 HTTP 消息都通过 IP 上的 TCP 进行发送。为便于参考，上面[方括弧]中的颜色显示了何种协议定义下面发现消息中的具体标头与值。

1.2.2 发现：搜索：M-SEARCH 形式请求

当一个控制点加入到网络中时，它应该采用以下格式的 M-SEARCH 方法发送多播请求。用*斜体*表示的值为实际值的占位符。

[M-SEARCH](#) * HTTP/1.1

HOST: [239.255.255.250:1900](#)
MAN:"[ssdp:discover](#)"
MX:*seconds to delay response*
ST:*search target*

（采用 M-SEARCH 方法发送的请求没有消息体，但请注意，消息与最后一个 HTTP 标头之间必须空一行。）

IP 信息包的 TTL 必须缺省为 4，并且可以配置。

下面的列表提供了上面出现的命令行与标头的详细信息。除特别注明外，所有标头值均要区分大小写。

命令行

M-SEARCH

SSDP 定义的搜索请求方法。

*

请求通用，不受具体资源的限制。必须是*。

HTTP/1.1

HTTP 版本。

标头

HOST

要求。由互联网编号分配组织（IANA）为 SSDP 保留的多播信道和端口。
必须是 [239.255.255.250:1900](#)。

MAN

要求。与 NTS 和 ST 标头不同，MAN 标头的值采用双引号。必须是
["ssdp:discover"](#)。

MX

要求。最长等待时间。设备响应在 0 和这个值之间随机选择响应延迟的值，这样可以为控制点响应平衡网络负载。如果大量设备需要响应或如果网络延迟过长，那么该值应该适当增大。具体由 UPnP 厂商决定。取整数。

ST

SSDP 要求使用的标头。搜索目标。必须是以下一种。(如上 `ssdp:alive NOTIFY` 中的 NT 标头)单一 URI。

[ssdp:all](#)

搜索所有设备和服务。

[UPnP:rootdevice](#)

只搜索根设备。

uuid:*device-UUID*

搜索特殊设备。设备 UUID 由 UPnP 厂商指定。

urn:[schemas-UPnP-org:device:deviceType:v](#)

搜索同类设备。设备类型与版本由 UPnP 论坛工作委员会定义。

urn:[schemas-UPnP-org:service:serviceType:v](#)

搜索同类服务。服务类型与版本由 UPnP 论坛工作委员会定义。

鉴于 UDP 的不可靠性，控制点应该多次发送每个 M-SEARCH 消息。因此，为了排除无法接收控制点 M-SEARCH 消息的可能性，设备应该定期重新发送其宣告信息（参见：如上述 `ssdp:alive NOTIFY` 中包含的 `CACHE-CONTROL` 标头）。

1.2.3 发现：搜索：响应

为了被找到，设备必须向发送查找请求的多播通道的源 IP 地址与端口发送响应信息。

传送给 M-SEARCH 的响应消息与宣告消息有意设计得类似，遵循上述 `ssdp:alive NOTIFY`（如上）相同的形式，处理将 `NT` 标头改做 `ST` 标头。响应必须采用以下格式发送：用斜体表示的值为实际值的占位符。

```
HTTP/1.1 200 OK
CACHE-CONTROL:max-age = seconds until advertisement expires
DATE:when response was generated
EXT:
LOCATION:URL for UPnP description for root device
SERVER:OS/version UPnP/1.0 product/version
ST:search target
USN:advertisement UUID
```

（采用 M-SEARCH 方法不包含响应消息体，但是请注意，消息与最后一个 HTTP 标头之间必须间隔一空行。）

（采用 IP 信息包响应请求时无需将 TTL 限定为 4。）

以下列表提供了上面出现的标头的详细信息。除特别说明外所有标头值区分大小写。

标头 (header)

CACHE-CONTROL

要求。必须带有 `max-age`，指代宣告有效持续时间。如果超过此时间间隔，控制点可以认为设备（或服务）不再可用。应大于 1800 秒（30 分钟）。具体由 UPnP 厂商决定。取整数。

DATE

推荐使用。响应生成时间。RFC1123 日期。

EXT

要求。确定 `MAN` 标头已经被理解。（只限标头，无值。）

LOCATION

要求。包含根设备 UPnP 描述的 URL 地址。在某些未受管理的网络中，URL 主机可能包含 IP 地址（与域名相对）。具体由 UPnP 厂商决定。单一 URL。

SERVER

要求。包含操作系统名称、操作系统版本、[UPnP/1.0](#)、产品名称以及产品版本等信息。具体由 UPnP 厂商决定。字符串。

ST

SSDP 要求使用的标头。搜索目标。单一 URI。如果请求中包含 ST 标头，必须是

[ssdp:all](#)

如果根设备带有 d 种嵌入式设备与 s 种嵌入服务且只有 K 种不同服务类型，那么其响应次数为 $3+2d+k$ 。ST 标头值必须与采用 `ssdp:alive` 的 NOTIFY 消息中的 NT 标头保持一致。（详见如上说明。）单一 URI。

UPnP:rootdevice

向根设备响应一次。必须是 [UPnP:rootdevice](#)。单一 URI。

uuid:*device-UUID*

向每种设备（根设备或嵌入式设备）响应一次。必须是 `uuid:device-UUID`。设备 UUID 由 UPnP 厂商指定。单一 URI。

urn:[schemas-UPnP-org:device:deviceType:v](#)

向每种设备（根设备或嵌入式设备）响应一次。必须是 `urn:schemas-UPnP-org:device:deviceType:v`。设备类型与版本由 UPnP 论坛工作委员会定义。

urn:[schemas-UPnP-org:service:serviceType:v](#)

向每种服务响应一次。必须是 `urn:schemas-UPnP-org:service:serviceType:v`。服务类型与版本由 UPnP 论坛工作委员会定义。

USN

SSDP 要求使用的标头。唯一服务名称。（详见以上 `ssdp:alive NOTIFY` 中 `USN` 标头要求值列表。）单一 `URI`。

鉴于 `UDP` 的不可靠性，每种响应设备都应发送多次。为了排除控制点未能接收响应的可能性，设备应该定期重新发送其宣告（参见：上述 `ssdp:alive NOTIFY` 中包含的 `CACHE-CONTROL` 标头）。

如果搜索请求出现错误，设备必须发送以下任意一种错误响应。

错误

`MAN header != ssdp:discover`

412 预处理失败。如果 `MAN` 标头值不等于 `ssdp:discover`，那么设备必须用 `HTTP` 错误 **412** 预处理失败做出响应。

其它错误可能被以下 `UPnP` 协议栈层返回。参考有关这些协议的文件了解详细信息。

1.3 发现参考

GENA

通用事件通知架构。IETF 草案。

HTTPMU

HTTPU

在 `UDP` 上实现 `HTTP` 协议多播以及 `HTTP` 协议单播。IETF 草案。

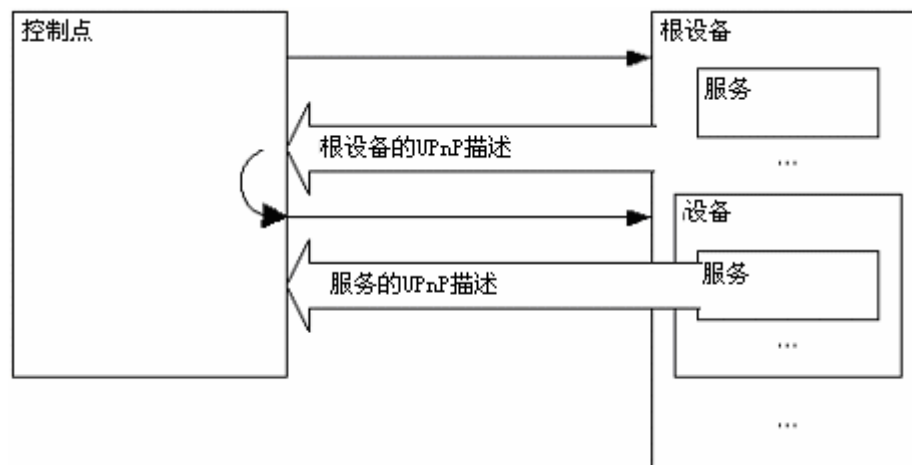
SSDP

简单服务发现协议。IETF 草案。

2. 描述

描述是 UPnP 网络的第二步。描述紧随寻址（第 0 步，设备获得网络地址）与发现（第 1 步，控制点发现感兴趣的设备）步骤之后。描述可以支持控制（第 3 步，控制点向设备发送命令）、事件触发（第 4 步，控制点监听设备状态的变化）以及展示（第 5 步，控制点显示设备的一个用户界面）。

在控制点发现了一个设备之后，控制点仍然对设备知之甚少——可能仅仅知道发现消息中的相关信息，如设备（或服务）的 UPnP 类型、设备的全球唯一标识符和设备 UPnP 描述的 URL 地址。为了让控制点更多的了解设备及其功能、或者与设备交互，控制点必须从发现消息中得到设备描述的 URL，并通过 URL 取得设备描述。



对于一个设备的 UPnP 描述一般分成两个逻辑部分：*设备描述*（描述所包含的物理与逻辑设备）以及一个或多个*服务描述*（描述设备对外暴露的能力）。UPnP 设备描述包括特定厂商、制造商信息，如模块名称和编号、序列号、制造商名称、特定厂商网站 URL 等（详细信息如下）。对于设备中的每种服务，设备描述都包含服务类型、名称、服务描述 URL、控制 URL 以及事件 URL。设备描述还包含所有嵌入式设备描述与 URL 地址集。本部分主要介绍 UPnP 设备描

述，控制、事件触发与展示部分将分别介绍如何使用控制 URL、事件 URL 与展示 URL。

注意一个物理设备可以包含多个逻辑设备。多个逻辑设备既可以是嵌入多个设备（服务）的单一根设备，也可以是多个根设备（可能没有嵌入式设备）。前者是一个有关根设备的 UPnP 设备描述，其中包含所有嵌入式设备描述。后者是多个 UPnP 设备描述，每个根设备对应一种描述。

UPnP 设备描述由 UPnP 厂商编写。采用 XML 句法，并且遵循标准 UPnP 设备模版。UPnP 设备模板由 UPnP 论坛工作委员会生成；该模板来源于 UPnP 模板语言——来自 XML 标准结构。本部分主要说明了 UPnP 设备描述格式、UPnP 设备模板、以及涵盖各种设备的 UPnP 模板语言等内容。

UPnP 服务描述包括一系列命令，或*动作*，服务的响应与每种动作的参数，或*变量*。服务描述还包括一系列变量。这些变量描述了服务运行时的状态，其中包括数据类型、取值范围和事件特性的描述。本部分主要说明了动作描述、变量、状态变量、以及这些变量的性质。事件触发部分主要说明事件的特性。

如同 UPnP 设备描述一样，UPnP 服务描述由 UPnP 厂商提供。描述采用 XML 句法，并且基于标准 UPnP 服务模版。UPnP 服务模板由 UPnP 论坛工作委员会生成；来源于 UPnP 模板语言，并在要求的时候使用人类语言将其适当扩大。UPnP 模板语言来源于 XML 标准结构。本部分主要说明了 UPnP 服务描述格式、UPnP 服务模板、人类语言的主要增加以及涵盖各种服务的 UPnP 模板语言等内容。

UPnP 厂商可以通过扩展服务（包括其它 UPnP 服务）或嵌入其它设备使其设备与众不同。一旦控制点取得特殊的设备描述，控制点可以使用这些新增特性用于控制和事件触发。设备和服务描述权威地记录了设备的实现。

取得 UPnP 设备描述非常简单：控制点在发现消息的 URL 上发出 HTTP GET 请求，然后设备返回设备描述。取得 UPnP 服务描述也是一个相似的流程，需要使用设备描述中的 URL。响应和请求的协议栈、方法、标头和正文详细解释如下：

只要来自设备的发现宣告没有过期，控制点就可以认为设备及其服务可用。只要设备及其服务可用，设备和服务描述就处于静止状态，因此，可以在任何地点获得。如果设备取消了其宣告，那么控制点就必须认为设备及其服务不再可用。如果设备需要改变其中一个描述，那么它必须取消已经发布的宣告消息并重新发布宣告信息。因此，如果设备再次出现在网络上时，控制点应该认为设备与服务描述有了变化。

和发现一样，描述在设备与控制点使用不同版本 UPnP 网络互操作性方面有着至关重要的作用。正如发现部分介绍的一样，UPnP 设备架构（此处定义）使用主版本和副版本来描述。副版本改进的部分必须兼容属于同一主版本的前副版本。主版本改进的部分不要求是前版的扩展集，并且并不保证向后兼容。版本信息用作发现消息的备份信息被包含在描述消息中。本部分主要说明了描述消息的格式。

剩余部分首先说明如何描述设备，详细阐述特定厂商信息、嵌入式设备以及控制 URL、事件触发 URL 和展示 URL。其次，说明 UPnP 设备模板。再次，说明如何描述服务，详细阐述动作、变量、状态变量、以及变量的性质。然后，说明 UPnP 服务模板和 UPnP 模板语言。最后，本部分详细说明控制点如何从设备取得设备和服务描述。

2.1 描述：设备描述

一个设备的 UPnP 描述包含多个特定厂商信息、所有嵌入式设备定义、设备展示 URL、以及所有服务列表，包括控制 URL 和事件触发 URL。除了定义非

标准设备之外，UPnP 厂商可以为标准设备添加新的嵌入式设备和服务。为了解释这一点，下面是针对实际元素和值的带有占位符（用斜体表示）的列表。其中一些占位符由 UPnP 论坛工作委员会（红色）或 UPnP 厂商（紫色）指定。对于非标准设备，所有占位符都由 UPnP 厂商具体指定。（在后面的参考信息中，由 UPnP 设备架构定义的元素以绿色表示。）紧随其后的是对于元素、属性和值的详细说明。

```
<?xml version="1.0"?>
<root xmlns="urn:schemas-upnp-org:device-1-0">
  <specVersion>
    <major>1</major>
    <minor>0</minor>
  </specVersion>
  <URLBase>base URL for all relative URLs</URLBase>
  <device>
    <deviceType>urn:schemas-upnp-org:device:deviceType:v</deviceType>
    <friendlyName>short user-friendly title</friendlyName>
    <manufacturer>manufacturer name</manufacturer>
    <manufacturerURL>URL to manufacturer site</manufacturerURL>
    <modelDescription>long user-friendly title</modelDescription>
    <modelName>model name</modelName>
    <modelName>model number</modelName>
    <modelURL>URL to model site</modelURL>
    <serialNumber>manufacturer's serial number</serialNumber>
    <UDN>uuid:UUID</UDN>
    <UPC>Universal Product Code</UPC>
    <iconList>
      <icon>
        <mimetype>image/format</mimetype>
        <width>horizontal pixels</width>
        <height>vertical pixels</height>
        <depth>color depth</depth>
        <url>URL to icon</url>
      </icon>
      XML to declare other icons, if any, go here
    </iconList>
    <serviceList>
      <service>
        <serviceType>urn:schemas-upnp-org:service:serviceType:v</serviceType>
        <serviceId>urn:upnp-org:serviceId:serviceID</serviceId>
        <SCPDURL>URL to service description</SCPDURL>
```



```

<controlURL>URL for control</controlURL>
<eventSubURL>URL for eventing</eventSubURL>
</service>
Declarations for other services defined by a UPnP Forum working committee (if any)
  go here
  Declarations for other services added by UPnP vendor (if any) go here
</serviceList>
<deviceList>
Description of embedded devices defined by a UPnP Forum working committee (if any)
  go here
  Description of embedded devices added by UPnP vendor (if any) go here
</deviceList>
<presentationURL>URL for presentation</presentationURL>
</device>
</root>

```

以下列出了上述列表中出现的元素、属性和值的详细信息。所有元素和属性都要区分大小写；HTTP 规定 URL 要区分大小写；其它值没有特别说明时不区分大小写。元素的顺序无关紧要。除非有特别说明：所要求的元素必须出现一次（无重复），推荐或可选元素最多只能出现一次。

xml

所有 XML 文件都要求：区分大小写。

root

要求。必须采用 urn:schemas-UPnP-org:device-1-0 作为 xmlns 属性值，参考 UPnP 模板语言（如下描述）。区分大小写。包含所有描述根设备的其它元素，即包含以下子元素：

specVersion

要求。包含以下子元素：

major

要求。UPnP 设备架构主版本。必须为 1。

minor

要求。UPnP 设备架构副版本。必须为 0。

URLBase

可选。定义基础 URL。用于构建完全合格的 URL。描述中其它地方出现的所有相关 URL 都附属于这个基础 URL。如果 URLBase 为空，或者没有提供，那么基础 URL 即为用以重新获得设备描述的 URL。具体由 UPnP 厂商决定。单个 URL。

device

要求。包含以下子元素：

deviceType

要求。UPnP 设备类型。

- 对于 UPnP 论坛工作委员会定义的标准设备，必须以 [urn:schemas-UPnP-org:device](#) 开头：后面带有设备类型后缀、冒号和整数设备版本（如上面列表所示）。
- 对于 UPnP 厂商指定的非标准设备，必须以 urn: 开头，带有厂商所有的 ICANN 域名，还有：[device:](#)，后面是设备类型后缀、冒号和整数设备版本，即 [urn:domain-name:device:deviceType:v](#)。

设备类型后缀由 UPnP 论坛工作委员会定义或由 UPnP 厂商来规定，必须不多于 64 个字符，版本后缀不计算在内，并且用冒号隔开。单一 URI。

friendlyName

要求。对于用户的简短描述。应该完成本地化（参见：[ACCEPT-/CONTENT-LANGUAGE](#) 标头）。具体由 UPnP 厂商决定。字符串。不得多于 64 个字符。

manufacturer

要求。制造商名称。可以进行本地化（参见：[ACCEPT-/CONTENT-LANGUAGE](#) 标头）。具体由 UPnP 厂商决定。字符串。不得多于 64 个字符。

manufacturerURL

可选。制造商网站。可以进行本地化（参见：**ACCEPT-/CONTENT-LANGUAGE** 标头）。可能与基础 URL 相关。具体由 UPnP 厂商决定。单一 URL。

modelDescription

推荐使用。对于用户的长篇描述。应该完成本地化（参见：**ACCEPT-/CONTENT-LANGUAGE** 标头）。具体由 UPnP 厂商决定。字符串。不得多于 128 个字符。

modelName

要求。型号名称。可以进行本地化（参见：**ACCEPT-/CONTENT-LANGUAGE** 标头）。具体由 UPnP 厂商决定。字符串。不得多于 32 个字符。

modelName

推荐使用。型号。可以进行本地化（参见：**ACCEPT-/CONTENT-LANGUAGE** 标头）。具体由 UPnP 厂商决定。字符串。不得多于 32 个字符。

modelURL

可选。型号网站。可以进行本地化（参见：**ACCEPT-/CONTENT-LANGUAGE** 标头）。可能与基础 URL 相关。具体由 UPnP 厂商决定。单一 URL。

serialNumber

推荐使用。序列号。可以进行本地化（参见：**ACCEPT-/CONTENT-LANGUAGE** 标头）。具体由 UPnP 厂商决定。字符串。应该少于 64 个字符。

UDN

要求。唯一设备名称。设备的全球唯一标识符，无论是根设备还是嵌入式设备。随着时间的推移，必须对具体设备例程保持一致（即必须在重启后继续保持不变）。必须与设备发现消息中的 **NT** 标头值相匹配。必须与所有发现消息中的 **USN** 标头前缀相匹配。

（“发现”章节将说明 NT 和 USN 标头。）必须以 **uuid:**开头：后面带有 UPnP 厂商指定的 UUID 后缀。单一 URI。

UPC

可选。通用产品代码。12 位全数字代码，用于确定销售包装。由统一编码委员会进行管理。具体由 UPnP 厂商决定。单一 UPC。

iconList

仅当设备有一个或多个图标时要求。具体由 UPnP 厂商决定。包含以下子元素：

icon

推荐使用。控制点 UI 中描述设备的图标。可以进行本地化（参见：**ACCEPT-/CONTENT-LANGUAGE** 标头）。下列每个尺寸推荐一个图标（宽度 x 高度 x 深度）：**16x16x1**、**16x16x8**、**32x32x1**、**32x32x8**、**48x48x1**、**48x48x8**。包含以下子元素：

mimetype

要求。图标的 MIME 类型（参见：**RFC2387**）。单一 MIME 图像类型。

width

要求。图标的水平尺寸，用像素、整数表示。

height

要求。图标的垂直尺寸，用像素、整数表示。

depth

要求。色彩位的数量，用像素、整数表示。

url

要求。面向图标图像的指针。（XML 不支持直接嵌入二进制数据。参见下面的说明。）通过 HTTP 取得。可能与基础 URL 相关。具体由 UPnP 厂商决定。单一 URL。

serviceList

要求。包含以下子元素：

service

要求。对于 UPnP 论坛工作委员会定义的每项服务要重复一次。如果 UPnP 厂商通过添加额外的标准 UPnP 服务来区分设备，则对于额外服务也要重复一次。包含以下子元素：

serviceType

要求。UPnP 服务类型。不得包含散列符（#，在 UTF-8 中为 23 Hex）。

- 对于由 UPnP 论坛工作委员会定义的标准服务类型，必须以 urn:[schemas-UPnP-org:service](#):开头，随后是服务类型后缀和整数服务版本（如上面列表所示）。
- 对于由 UPnP 厂商指定的非标准服务类型，必须以 urn:开头，随后是厂商所有的 ICANN 域名，然后是:[service](#):，接着是服务类型后缀、冒号和整数服务版本，即 urn:*domain-name*:[service](#):*serviceType*:*v*。

由 UPnP 论坛工作委员会定义或 UPnP 厂商具体指定的服务类型后缀不得多于 64 个字符，版本后缀不计算在内，并且用冒号隔开。单一 URI。

serviceId

要求。服务标识符。在这一设备描述中必须保持唯一性。

- 对于由 UPnP 论坛工作委员会定义的标准服务，必须以 urn:[UPnP-org:serviceId](#):开头，后面是服务 ID 后缀（如上面列表所示）。（注意在此时要使用 [UPnP-org](#) 来代替 [schemas-UPnP-org](#)，因为 XML 模式并没有针对每个服务 ID 进行定义。）
- 对于由 UPnP 厂商指定的非标准服务，必须以 urn:开头，随后是厂商拥有的 ICANN 域名，然后是:[serviceId](#):，接着是服务 ID 后缀，即 urn:*domain-name*:[serviceId](#):*serviceID*。

服务 ID 后缀由 UPnP 论坛工作委员会定义或由 UPnP 厂商指定，不得超过 64 个字符。单一 URL。

SCPDURL

要求。服务描述的 URL (以前的服务控制协议定义 URL)。(参见：下面关于服务描述的章节。)可能与基础 URL 相关。具体由 UPnP 厂商决定。单一 URL。

controlURL

要求。控制的 URL (参见：关于控制的章节)。可能与基础 URL 相关。具体由 UPnP 厂商决定。单一 URL。

eventSubURL

要求。事件的 URL (参见：关于事件的章节)。可能与基础 URL 相关。在设备内部可能是唯一的；不会有二个服务拥有相同的事件 URL。如果服务没有事件变量，则它应该没有事件 (参见：关于事件的章节)；如果服务没有事件，则该元素必须展示出来，但是应该为空，即 <eventSubURL></eventSubURL>。具体由 UPnP 厂商决定。单一 URL。

deviceList

仅当根设备带有嵌入式设备时要求。包含以下子元素：

device

要求。对于 UPnP 论坛工作委员会定义的每项嵌入式设备要重复一次。如果 UPnP 厂商通过嵌入其它 UPnP 设备来区分设备，则对于每个嵌入式设备要重复一次。根据上面的定义，包含面向根子元素设备的子元素。

presentationURL

推荐使用。设备展示的 URL (参见：关于展示的章节)。可能与基础 URL 相关。具体由 UPnP 厂商决定。单一 URL。

为了获得未来可伸缩性，在处理上述 XML 时要遵守灵活 XML 处理框架（FXPP），设备和控制点必须忽略：(a) 任何未知元素及其子元素或内容，以及(b)任何未知属性及其值。

注意“和”字符（&，在 UTF-8 中为 0x26）不允许出现在 XML 中。如果需要作为 XML 元素（如一个 URL）的值的一部分，“和”符号必须转换为&（HTML）或%26（URL 转义码）。

XML 不支持直接嵌入的二进制数据，如 UPnP 设备描述中的图标。二进制数据可以转换为文本（从而嵌入 XML 中）使用 bin.base64（用于二进制数据的一种 MIME 型的基础 64 编码）或 bin.hex（代表八位字节的十六进制数字）的 XML 数据类型。此外，通过在 XML 中嵌入一个 URL 并传输数据以响应单独的 HTTP 请求，数据可以进行间接传递并保持原样；UPnP 设备描述中的图标将以后一种方式进行传送。

由 UPnP 论坛工作委员会进行标准化的设备有一个完整版本。设备的每一个以后的版本必须包括以前的版本，也就是说，与设备的早期版本相比，它必须包括相同或以后版本的所有嵌入式设备和服务。尽管以后的设备版本越来越大，但是一个设备的所有版本的 UPnP 设备类型仍然保持一致。

2.2 描述：UPnP 设备模板

上面的列表也显示出 UPnP 设备描述与 UPnP 设备模板之间的关系。正如上面所说明的，UPnP 设备描述由 UPnP 厂商按照 UPnP 设备模板用 XML 来编写。UPnP 设备模板由 UPnP 论坛工作委员会进行制作，作为一种对设备进行标准化的方式。

根据预留位置的适当的规范，上面的列表可能是一个 UPnP 设备模板或一个 UPnP 设备描述。注意一些由 UPnP 论坛工作委员会定义的占位符(用红色表示)，

即 UPnP 设备类型标识符，要求的 UPnP 服务，以及要求的 UPnP 嵌入式设备（如果有）。如果这些内容都已经定义，则列表就是 UPnP 设备模板，成为针对这一设备类型的标准。UPnP 设备模板是 UPnP 论坛工作委员会的主要成果之一。

进一步说，上面列表中剩余的占位符将由 UPnP 厂商来规定（以紫色表示），即具体厂商的信息。如果这些占位符已经指定（及其它），则列表就是 UPnP 设备描述，适合于提供给控制点以便支持控制、事件和展示。

换句话说，UPnP 设备模板用来定义整体设备类型，而每个 UPnP 设备描述则是将厂商具体信息添加到模板中。第一个由 UPnP 论坛工作委员会创建；以后的由 UPnP 厂商来制作。

2.3 描述：服务描述

关于服务的 UPnP 描述定义了动作及其变量，还有状态变量及其数据类型、范围和事件特征。

每个服务可能有零或多个动作。每个动作可能有零或更多变量。这些变量的任何组合可能是输入或输出参数。如果一个动作有一个或多个输出变量，那么这些变量中的一个可能被标记为返回值。每个变量都应该对应一个状态变量。这一直接处理编程模式可以加强简单性。

每个服务必须有一个或多个状态变量。

除了定义非标准服务，UPnP 厂商可以为标准设备添加新的动作和服务。

为了解释这几项，下面是针对实际元素和值的带有占位符（用斜体表示）的列表。对于标准的 UPnP 服务，其中一些占位符由 UPnP 论坛工作委员会（用

[红色](#)表示) 定义或由 UPnP 厂商 ([紫色](#)) 指定。对于非标准服务, 所有占位符都由 UPnP 厂商具体指定。(在后面的参考信息中, 由 UPnP 设备架构定义的元素以[绿色](#)表示。) 紧随其后的是对于元素、属性和值的详细说明。

```
<?xml version="1.0"?>
<scpd xmlns="urn:schemas-UPnP-org:service-1-0">
  <specVersion>
    <major>1</major>
    <minor>0</minor>
  </specVersion>
  <actionList>
    <action>
      <name>actionName</name>
      <argumentList>
        <argument>
          <name>formalParameterName</name>
          <direction>in xor out</direction>
          <retval />
          <relatedStateVariable>stateVariableName</relatedStateVariable>
        </argument>
        Declarations for other arguments defined by UPnP Forum working committee \(if any\)
        go here
      </argumentList>
    </action>
    Declarations for other actions defined by UPnP Forum working committee \(if any\)
    go here
    Declarations for other actions added by UPnP vendor \(if any\) go here
  </actionList>
  <serviceStateTable>
    <stateVariable sendEvents="yes">
      <name>variableName</name>
      <dataType>variable data type</dataType>
      <defaultValue>default value</defaultValue>
      <allowedValueList>
        <allowedValue>enumerated value</allowedValue>
        Other allowed values defined by UPnP Forum working committee \(if any\) go here
      </allowedValueList>
    </stateVariable>
    <stateVariable sendEvents="yes">
      <name>variableName</name>
      <dataType>variable data type</dataType>
      <defaultValue>default value</defaultValue>
      <allowedValueRange>
```

```
<minimum>minimum value</minimum>
<maximum>maximum value</maximum>
<step>increment value</step>
</allowedValueRange>
</stateVariable>
Declarations for other state variables defined by UPnP Forum working committee
(if any) go here
Declarations for other state variables added by UPnP vendor (if any) go here
</serviceStateTable>
</scpd>
```

以下列出了上述列表中出现的元素、属性和值的详细信息。所有元素和属性均区分大小写；除特别注明外，值不区分大小写。除特别注明外，元素的顺序无关紧要。除特别注明外，所要求的元素必须出现一次（无重复），推荐或可选元素最多只能出现一次。

xml

要求所有 XML 文件采用。区分大小写。

scpd

要求。必须采用 [urn:schemas-UPnP-org:service-1-0](#) 作为 xmlns 属性值，参考 UPnP 模板语言（如下描述）。区分大小写。包含所有描述服务的其它元素，即包含以下子元素：

specVersion

要求。包含以下子元素：

major

要求。UPnP 设备架构主版本。必须为 1。

minor

要求。UPnP 设备架构副版本。必须为 0。

actionList

仅当服务带有动作时要求。（每个服务都可以有不少于 0 个动作。）

包含以下子元素：

action

要求。对于 UPnP 论坛工作委员会定义的每项动作要重复一次。如果 UPnP 厂商通过添加额外的动作来区分服务，则对于每个额外的动作都要重复一次。包含以下子元素：

name

要求。动作的名称。不得包含连字符（-，在 UTF-8 中为 2D Hex）和散列符（#，在 UTF-8 中为 23 Hex）。

- 对于 UPnP 论坛工作委员会定义的标准动作，不得以 **X** 和 **A** 开头。
- 对于 UPnP 厂商指定的非标准动作和标准服务的添加项目，必须使用 **X** 开头。

字符串。应该不超过 32 个字符。

argumentList

仅当为动作定义参数时要求。（每个动作都可以有不少于 0 个参数。）包含以下子元素：

argument

要求。为每个参数重复一次。包含以下子元素：

name

要求。正式参数的名称。应该是动作所影响的模型的状态变量的名称。不得包含连字符（-，在 UTF-8 中为 2D Hex）。字符串。应该不超过 32 个字符。

direction

要求。无论变量是输入还是输出参数。必须采用 in xor out 格式。任何 in 变量必须列在任何 out 变量的前面。

retval

可选。最多确定一个 out 变量作为返回值。如果包括，必须是第一个 out 变量。（只限元素，无值。）

relatedStateVariable

要求。必须是一个状态变量的名称。

serviceStateTable

要求。（每个服务必须有不少于 0 状态变量。）包含以下子元素：

stateVariable

要求。对于 UPnP 论坛工作委员会定义的每个状态变量都要重复一次。如果 UPnP 厂商通过添加额外的状态变量来区分服务，则对于每个额外变量都要重复一次。**sendEvents** 属性将定义当这一状态变量的值发生变化时是否生成事件消息；非事件状态变量采用 **sendEvents= “no”**，缺省为 **sendEvents= “yes”**。包含以下子元素：

name

要求。状态变量的名称。不得包含连字符（-，在 UTF-8 中为 2D Hex）。

- 对于 UPnP 论坛工作委员会定义的标准变量，不得以 **X** 和 **A** 开头。
- 对于 UPnP 厂商指定的非标准变量和标准服务的添加项目，必须使用 **X** 开头。

字符串。应该不超过 32 个字符。

dataType

要求。与 XML 模式，部分 2：数据类型中定义的数据类型相同。由 UPnP 论坛工作委员会定义的类型用于标准状态变量；由 UPnP 厂商指定的类型用于扩展。必须是以下值中的一种：

ui1

无符号的 1 字节 int。与没有引导标志（leading sign）的 int 格式相同。

ui2

无符号的 2 字节 int。与没有引导标志（leading sign）的 int 格式相同。

ui4

无符号的 4 字节 int。与没有引导标志（leading sign）的 int 格式相同。

i1

1 字节 int。与 int 格式相同。

i2

2 字节 int。与 int 格式相同。

i4

4 字节 int。与 int 格式相同。必须在 -2147483648 和 2147483647 之间。

int

固定点，整数。可能有引导标志 (leading sign)。可能有引导零 (leading zero)。(没有当前符号。)(小数左侧没有数字组，如没有逗号。)

r4

4 字节浮点。与浮点格式相同。必须在 $3.40282347E+38$ 到 $1.17549435E-38$ 之间。

r8

8 字节浮点。与浮点格式相同。负值必须在 $-1.79769313486232E308$ 和 $-4.94065645841247E-324$ 之间，正值必须在 $4.94065645841247E-324$ 和 $1.79769313486232E308$ 之间，也就是说 IEEE 64 位 (8 字节) 加倍。

number

与 r8 相同。

fixed.14.4

与 r8 相同，但是小数点左侧不超过 14 个数字，右侧不超过 4 个数字。

float

浮点数字。尾数 (小数的左侧) 和/或指数可能有引导标志 (leading sign)。尾数和/或指数可能有引导零 (leading zero)。尾数中的小数符号是一个句点，也就是说，尾数的整个数字由句点分割成分数 (fractional digit)。尾数由

E 与指数分开。（无当前符号。）（尾数中没有数字组，如没有句点。）

char

统一字符字符串。一个字符长。

string

统一字符字符串。没有长度限制。

date

ISO 8601 格式子集中的日期没有时间数据。

dateTime

ISO 8601 格式的日期，带有可选时间但是没有时区。

dateTime.tz

ISO 8601 格式的日期，带有可选时间但是没有可选时区。

time

ISO 8601 格式子集中的时间，没有日期和时区。

time.tz

ISO 8601 格式子集中的时间，带有可选时区但是没有日期。

boolean

0, false, 或者用 no 表示 false; 1, true, 或者用 yes 表示 true。

bin.base64

MIME 型 Base64 编码二进制 BLOB。取 3 个字节，将其分为 4 部分，每 6 位形成一组，与一个 8 位字节相匹配。（3 个 8 位字节编码为 4。）没有大小限制。

bin.hex

十六进制数字代表 8 位字节。将每个半位元组作为一个十六进制数字，编码为一个独立的字节。（1 个 8 位字节编码为 2。）没有大小限制。

uri

统一资源标识符。

uuid

全球唯一 ID。十六进制数字代表 8 位字节。可选的嵌入式连字符可以忽略。

defaultValue

推荐使用。期望的初值。由 UPnP 论坛工作委员会定义或委托给 UPnP 厂商。必须匹配数据类型。必须满足 **allowedValueList** 或 **allowedValueRange** 限制。

allowedValueList

推荐使用。列举合法的字符串值。禁止数据类型而不是字符串。最多可以规定 **allowedValueRange** 和 **allowedValueList** 其中之一。子元素进行了排序（例如，参见 **NEXT_STRING_BOUNDED**）。包含以下子元素：

allowedValue

要求。字符串变量的合法值。由 UPnP 论坛工作委员会定义的类型用于标准状态变量；由 UPnP 厂商指定的类型用于扩展。应该不超过 32 个字符。

allowedValueRange

推荐使用。为合法数值定义范围；为数值定义解决方法。仅面对数字数据类型进行定义。最多可以规定 **allowedValueRange** 和 **allowedValueList** 其中之一。包含以下子元素：

minimum

要求。包括下限。由 UPnP 论坛工作委员会定义或委托给 UPnP 厂商。单一数值。

maximum

要求。包括上限。由 UPnP 论坛工作委员会定义或委托给 UPnP 厂商。单一数值。

step

推荐使用。增量运算的大小，也就是操作 $v = v + s$ 中 s 的值。由 UPnP 论坛工作委员会定义或委托给 UPnP 厂商。单一数值。

对于未来的可伸缩性，在处理上述 XML 时，要遵守灵活的 XML 处理框架（FXPP），设备和控制点必须忽略：**(a)** 任何未知元素及其子元素或内容，以及**(b)**任何未知属性及其值。

注意“和”字符（&，在 UTF-8 中为 26 Hex）不允许出现在 XML 中。如果需要作为 XML 元素（如一个 URL）的值的一部分，“和”符号必须转换为&（HTML）或%26（URL 转义码）。

注意对于一个服务而言在逻辑上依然可能虽然没有动作但是有状态变量和活动；虽然不太可能发生，但是这样的服务将成为独立的信息来源。然而，没有状态变量的服务是被禁止的。

与设备描述不同，服务描述及相关值应该不使用特定位置的值；其中包括服务描述、动作变量值、以及状态变量值。与此相反，大多数动作变量和状态变量都应该使用与位置相独立的值表示方法；应用应该进行转换和/或格式化，将信息从标准形式变为适合场所的正确语言和/或格式。例如，日期用与位置相对立的格式（ISO 8601）来表示，用整数表示，没有场所独特的格式（如没有货币符号、没有数字组）。串值应该用标准的‘场所’或统一方法来表示。带有 allowedValueList 的变量应该使用 UPnP 标准语言的令牌值，而且不反映要显示在场所化用户界面上的字符串。

然而，一些情况下动作的行为取决于具体的场所。在这种情况下，变量应定义用于指示场所，或者可以使用与 ACCEPT-/CONTENT-LANGUAGE 标头相同的编码方法（RFC1766）。如果有多个独立于场所之外的动作，则服务可以包

括一个动作以便设置一个状态变量来指示地点，而且也不需要将地点标识符分别传递给每个动作。

由 UPnP 论坛工作委员会进行标准化的服务有一个完整版本。服务每一个以后的版本必须包括以前的版本，也就是说，它必须准确地包括服务的早期版本定义的所有动作和状态变量。尽管以后的服务版本越来越大，但是一个服务的所有版本的 UPnP 服务类型仍然保持一致。

2.4 描述：UPnP 服务模板

上面的列表也显示出 UPnP 服务描述与 UPnP 服务模板之间的关系。正如上面所说明的，UPnP 对于服务的描述由 UPnP 厂商按照 UPnP 服务模板用 XML 来编写。UPnP 服务模板由 UPnP 论坛工作委员会进行制作，作为一种对设备进行标准化的方式。

根据占位符的适当的规范，上面的列表可能是一个 UPnP 服务模板或一个 UPnP 服务描述。注意一些由 UPnP 论坛工作委员会定义的占位符(用 **红色**表示)，即动作及其参数、状态及其数据类型、范围和事件特征。如果这些内容都已经指定，则上述列表就是 UPnP 服务模板，并成为针对这一服务类型的标准。除 UPnP 设备模板（参见：关于描述的章节）之外，UPnP 服务模板也是 UPnP 论坛工作委员会的主要成果之一。

进一步说，上面列表中剩余的占位符将由 UPnP 厂商来规定(以 **紫色**表示)，即额外的具体厂商的动作和状态变量。如果这些占位符已经指定（与其它的内容一样），则上述列表可能是 UPnP 服务描述，适合于一个设备内的服务效果控制。

换句话说，UPnP 服务模板用来定义整体服务类型，而每个 UPnP 服务描述则是将厂商具体内容添加到模板中。第一个由 UPnP 论坛工作委员会创建；以后的由 UPnP 厂商来制作。

2.5 描述：非标准厂商扩展

如上所述，UPnP 厂商可能通过包括额外的服务以及嵌入式设备来区分他们的设备并扩展标准设备。同样地，UPnP 厂商也可以通过包括额外的动作或状态变量来扩展标准服务。UPnP 厂商不得通过修改标准化的 `allowedValueList` 来扩展标准服务。下表中列出针对每种类型的命名惯例，并给出详细说明。

扩展类型	标准	非标准
设备类型	urn: <u>schemas-UPnP-org:device:deviceType:v</u>	urn: <i>domain-name:device:deviceType:v</i>
服务类型	urn : <u>schemas-UPnP-org:service:serviceType:v</u>	urn: <i>domain-name:service:serviceType:v</i>
服务 ID	urn: <u>UPnP-org:serviceId:serviceID</u>	urn: <i>domain-name:serviceId:serviceID</i>
动作名称	不要以 <u>X</u> 或 <u>A</u> 开头。	用 <u>X</u> 开头。
状态变量名称	不要以 <u>X</u> 或 <u>A</u> 开头。	用 <u>X</u> 开头。
设备或服务描述中的 XML 元素	由 UPnP 模板语言进行定义。	由 XML 名称空间确定范围并位于以 <u>X</u> 开头的元素内的任意 XML。
设备或服务描述中的 XML 属性	由 UPnP 模板语言进行定义。	由 XML 名称空间确定范围并以 <u>X</u> 开头的任意属性。

正如上表的最后二行所说明的一样，UPnP 厂商也可以向设备或服务描述添加非标准 XML。每个添加项必须在厂商提供的 XML 名称空间的范围内。任意 XML 必须包括以 X_开头的元素，该元素必须是包含子元素的标准元素的子元素。非标准属性也可以添加到标准元素中，但是这些属性必须在 XML 名称范围内而且以 X_开头。

为了说明这一点，下面是针对实际元素和值的带有占位符（用斜体表示）的列表。其中一些占位符将由 UPnP 厂商（紫色）来指定，另外一些由 UPnP 设备架构（绿色）进行定义。

```
<RootStandardElement xmlns="urn:schemas-UPnP-org:device-1-0"  
  xmlns:n="domain-name:schema-name">  
  other XML  
  <AnyStandardElement n:X_VendorAttribute="arbitrary string value">  
    other XML  
  </AnyStandardElement>  
  other XML  
</RootStandardElement>
```

RootStandardElement

要求。一个标准的根元素。xmlns 属性定义了名称空间，在这种情况下，一个标准 UPnP 名称空间和一个非标准名称空间（带有前缀 *n*）。

- 对于设备描述，必须是 **root**。
- 对于服务描述，必须是 **scpd**。

AnyStandardElement

要求。任何标准元素，无论是根元素还是其它元素，无论是文本内容还是元素本身，必须已经成为标准的设备或服务描述的组成部分。

X_VendorAttribute 必须以 **X** 开头。（前缀 **A** 已经保留。）可能有一个任意的字符串值。

```
<EltOnlyStandardElement n:X_VendorAttribute="vendor value">  
<n:X_VendorElement xmlns:n="domain-name:schema-name">  
  arbitrary XML  
</n:X_VendorElement>  
</EltOnlyStandardElement>
```

EltOnlyStandardElement

要求。元素只包含元素的内容。必须已经成为标准的设备或服务描述的组成部分。

- 对于设备描述,必须是下列之一:[root](#)、[specVersion](#)、[device](#)、[iconList](#)、[icon](#)、[serviceList](#)、[service](#) 和/或 [deviceList](#)。
- 对于服务描述,必须是下列之一: [scpd](#)、[actionList](#)、[action](#)、[argumentList](#)、[argument](#)、[serviceStateTable](#)、[stateVariable](#)、[allowedValueList](#) 和/或 [allowedValueRange](#)。

X *VendorElement*

要求。必须用 X 开头。(前缀 A 已经保留。) 必须有面向 `xmlns` 属性的值。可以包括任意 XML。

根据灵活的 XML 处理框架 (FXPP) 的规定, 不了解这些 XML 添加项的控制点必须将其忽略。

2.6 描述: 用于设备的 UPnP 模板语言

上面的段落说明了 UPnP 设备描述并说明了一个描述如何从 UPnP 设备模板进行例示。根据说明, UPnP 设备模板由 UPnP 论坛工作委员会进行制作, 这些模板均源自于 UPnP 模板语言。这一模板语言可以为设备和服务定义有效模板。下面是与设备相关的这一语言的列表和说明。

UPnP 模板语言用 XML 句法编写, 源自 XML 模式 (部分 1: 结构, 部分 2: 数据类型)。XML 模式提供了一套 XML 解释, 以便表达语言概念, 如要求与可选元素、元素嵌套以及取值的数据类型等 (还有其它与此处无关的属性)。UPnP 模板语言使用这些 XML 模式解释来定义 `specVersion`、`URLBase`、`deviceType` 以及上面详细介绍的元素。由于 UPnP 模板语言使用另外一种精确的语言来解释, 所以它是无歧义的。同时因为 UPnP 模板语言、UPnP 设备模板和 UPnP

设备描述全部为机器可读型，所以自动工具可以自动进行检查以确保后两者拥有所有要求的元素，在正确的嵌套内，而且有正确数据类型的值。

下面是面向设备的 UPnP 模板语言，在此由 UPnP 设备架构进行定义。它定义的元素用于 UPnP 设备模板中；它们在这里用绿色表示，在上面的列表中同样用绿色表示。下面是定义这些元素的地方；上面是使用它们的地方。

紧随其后的是对于 XML 模式元素、属性和所用值的简单说明。章节尾部关于 XML 模式的参考中还有更多详细内容。

用于设备的 UPnP 模板语言

```
<?xml version="1.0"?>
<Schema name="device-1-0"
  xmlns="urn:schemas-microsoft-com:xml-data"
  xmlns:dt="urn:schemas-microsoft-com:datatypes">
  <ElementType name="root" content="eltOnly">
    <element type="specVersion" />
    <element type="URLBase" minOccurs="0" maxOccurs="1" />
    <element type="device" />
  </ElementType>
  <ElementType name="specVersion" content="eltOnly">
    <element type="major" />
    <element type="minor" />
  </ElementType>
  <ElementType name="major" dt:type="int" content="textOnly" />
  <ElementType name="minor" dt:type="int" content="textOnly" />
  <ElementType name="URLBase" dt:type="uri" content="textOnly" />
  <ElementType name="device" content="eltOnly">
    <element type="deviceType" />
    <element type="friendlyName" />
    <element type="manufacturer" />
    <element type="manufacturerURL" minOccurs="0" maxOccurs="1" />
    <element type="modelDescription" minOccurs="0" maxOccurs="1" />
    <element type="modelName" />
    <element type="modelNumber" minOccurs="0" maxOccurs="1" />
    <element type="modelURL" minOccurs="0" maxOccurs="1" />
    <element type="serialNumber" minOccurs="0" maxOccurs="1" />
  </ElementType>
</Schema>
```

```
<element type="UDN" />
<element type="UPC" minOccurs="0" maxOccurs="1" />
<element type="iconList" minOccurs="0" maxOccurs="1" />
<element type="serviceList" />
<element type="deviceList" minOccurs="0" maxOccurs="1" />
<element type="presentationURL" minOccurs="0" maxOccurs="1" />
</ElementType>
<ElementType name="deviceType" dt:type="uri" content="textOnly" />
<ElementType name="friendlyName" dt:type="string" content="textOnly" />
<ElementType name="manufacturer" dt:type="string" content="textOnly" />
<ElementType name="manufacturerURL" dt:type="uri" content="textOnly" />
<ElementType name="modelDescription" dt:type="string" content="textOnly" />
<ElementType name="modelName" dt:type="string" content="textOnly" />
<ElementType name="modelNumber" dt:type="string" content="textOnly" />
<ElementType name="modelURL" dt:type="uri" content="textOnly" />
<ElementType name="serialNumber" dt:type="string" content="textOnly" />
<ElementType name="UDN" dt:type="uri" content="textOnly" />
<ElementType name="UPC" dt:type="string" content="textOnly" />
<ElementType name="iconList" content="eltOnly">
<element type="icon" minOccurs="1" maxOccurs="*" />
</ElementType>
<ElementType name="icon" content="eltOnly">
<element type="mimetype" />
<element type="width" />
<element type="height" />
<element type="depth" />
<element type="url" />
</ElementType>
<ElementType name="mimetype" dt:type="string" content="textOnly" />
<ElementType name="width" dt:type="int" content="textOnly" />
<ElementType name="height" dt:type="int" content="textOnly" />
<ElementType name="depth" dt:type="int" content="textOnly" />
<ElementType name="url" dt:type="uri" content="textOnly" />
<ElementType name="serviceList" content="eltOnly">
<element type="service" minOccurs="1" maxOccurs="*" />
</ElementType>
<ElementType name="service" content="eltOnly">
<element type="serviceType" />
<element type="serviceId" />
<element type="SCPDURL" />
<element type="controlURL" />
<element type="eventSubURL" />
</ElementType>
<ElementType name="serviceType" dt:type="uri" content="textOnly" />
```

```
<ElementType name="serviceld" dt:type="uri" content="textOnly" />
<ElementType name="SCPDURL" dt:type="uri" content="textOnly" />
<ElementType name="controlURL" dt:type="uri" content="textOnly" />
<ElementType name="eventSubURL" dt:type="uri" content="textOnly" />
<ElementType name="deviceList" content="eltOnly">
  <element type="device" minOccurs="1" maxOccurs="*" />
</ElementType>
<ElementType name="presentationURL" dt:type="uri" content="textOnly" />
</Schema>
```

ElementType

用全新的衍生语言定义一个元素。名称属性定义了元素的名称。**dt:type** 属性用全新的衍生语言定义了元素值的数据类型。

element

参考一个元素的目的是宣布嵌套。**minOccurs** 属性定义了元素必须发生的最少次数；缺省为 **minOccurs = 1**；可选元素有 **minOccurs = 0**。**maxOccurs** 属性定义了元素必须发生的最多次数；缺省为 **maxOccurs = 1**；元素显示一次或多次使用 **maxOccurs = ***。

2.7 描述：用于服务的 UPnP 模板语言

上面的段落说明了 UPnP 服务描述并解释了一个描述如何从 UPnP 设备模板进行例示。与 UPnP 设备模板相同，UPnP 设备模板也是由 UPnP 论坛工作委员会进行制作，这些模板均源自于 UPnP 模板语言。这一模板语言可以为设备和服务定义有效模板。像上面说明的一样，UPnP 模板语言用 XML 句法编写，源自 XML 模式（部分 1：结构，部分 2：数据类型）。下面是与服务相关的语言列表。它定义的元素用于 UPnP 服务模板中；它们在这里用绿色表示，在上面的列表中同样用绿色表示。下面是定义这些元素的地方；上面是使用它们的地方。

紧随其后的是对于 XML 模式元素、属性和所用值的简单说明。章节尾部关于 XML 模式的参考中还有更多详细内容。

用于服务的 UPnP 模板语言

```
<?xml version="1.0"?>
<Schema name="service-1-0"
  xmlns="urn:schemas-microsoft-com:xml-data"
  xmlns:dt="urn:schemas-microsoft-com:datatypes">
  <ElementType name="scpd" content="eltOnly">
  <element type="specVersion" />
  <element type="actionList" minOccurs="0" maxOccurs="1" />
  <element type="serviceStateTable" />
  </ElementType>
  <ElementType name="specVersion" content="eltOnly">
  <element type="major" />
  <element type="minor" />
  </ElementType>
  <ElementType name="major" dt:type="int" content="textOnly" />
  <ElementType name="minor" dt:type="int" content="textOnly" />
  <ElementType name="actionList" content="eltOnly">
  <element type="action" minOccurs="1" maxOccurs="*" />
  </ElementType>
  <ElementType name="action" content="eltOnly">
  <element type="name" />
  <element type="argumentList" minOccurs="0" maxOccurs="1" />
  </ElementType>
  <ElementType name="name" dt:type="string" content="textOnly" />
  <ElementType name="argumentList" content="eltOnly">
  <element type="argument" minOccurs="1" maxOccurs="*" />
  </ElementType>
  <ElementType name="argument" content="eltOnly">
  <element type="name" />
  <element type="direction" />
  <element type="retval" minOccurs="0" maxOccurs="1" />
  <element type="relatedStateVariable" />
  </ElementType>
  <ElementType name="direction" dt:type="string" content="textOnly" />
  <ElementType name="retval" content="empty" />
  <ElementType name="relatedStateVariable" dt:type="string" content="textOnly" />
  <ElementType name="serviceStateTable" content="eltOnly">
  <element type="stateVariable" minOccurs="1" maxOccurs="*" />
```



```

</ElementType>
<ElementType name="stateVariable" content="eltOnly">
  <element type="name" />
  <element type="dataType" />
  <element type="defaultValue" minOccurs="0" maxOccurs="1" />
  <group minOccurs="0" maxOccurs="1" order="one">
    <element type="allowedValueList" />
    <element type="allowedValueRange" />
  </group>
  <AttributeType name="sendEvents" />
  <attribute default="yes" type="sendEvents" required="no" />
</ElementType>
<ElementType name="dataType" dt:type="string" content="textOnly" />
<ElementType name="defaultValue" dt:type="string" content="textOnly" />
<ElementType name="allowedValueList" content="eltOnly">
  <element type="allowedValue" minOccurs="1" maxOccurs="*" />
</ElementType>
<ElementType name="allowedValue" content="textOnly" />
<ElementType name="allowedValueRange" content="eltOnly">
  <element type="minimum" />
  <element type="maximum" />
  <element type="step" minOccurs="0" maxOccurs="1" />
</ElementType>
<ElementType name="minimum" dt:type="number" content="textOnly" />
<ElementType name="maximum" dt:type="number" content="textOnly" />
<ElementType name="step" dt:type="number" content="textOnly" />
</Schema>

```

attribute

参考新的衍生语言中的属性，旨在宣布可能出现元素的地点。与任何 XML 元素类似，属性元素可能有它自己的属性。使用这一元素内要求的属性指示为是否属性必须出现；可选属性表示为 **required = no**。

AttributeType

用全新的衍生语言定义一个新的属性。与任何 XML 元素类似，属性元素可能有它自己的属性。使用这一元素内的名称属性来定义属性的名称，因为它将用在衍生语言中。

element

参考一个元素的目的是宣布嵌套。**minOccurs** 属性定义了元素必须发生的最少次数；缺省为 **minOccurs = 1**；可选元素有 **minOccurs = 0**。

`maxOccurs` 属性定义了元素必须发生的最大次数；缺省为 `maxOccurs = 1`；元素显示一次或多次使用 `maxOccurs = *`。

ElementType

用全新的衍生语言定义一个元素。名称属性定义元素名称。`dt:type` 属性用全新的衍生语言定义元素值的数据类型。型号属性指示使用全新的衍生语言的元素是否包含此处没有明确规定的元素；当只有以前规定的元素可以使用时，`model = closed`。内容属性指示包含哪些内容；只包含其它元素的元素应该用 `content = eltOnly`；只包含字符串的元素应该用 `content = textOnly`。

group

将内容组织到一个组中以便规定顺序。`minOccurs` 属性定义了组必须发生的最少次数。`maxOccurs` 属性定义了组必须发生的最大次数。`order` 属性限制了元素的顺序；当最多只能出现一个元素时，`order = one`。

2.8 描述：增加 UPnP 模板语言

服务的一些属性在 XML 模式形式中很难捕捉到。尤其是，在描述动作对于状态变量的效果时十分有用。这一过程信息很难用 XML 这样的陈述式语言来描述，因此下面是推荐 UPnP 论坛工作委员会使用的词汇表，在定义服务动作时使用或由 UPnP 厂商在他们希望记录额外动作的效果时使用。

ASSIGN (*v*, *a*)

变量 *v* 成为变量 *a* 的值，即 $v = a$ 。*v* 和 *a* 必须是相同的数据类型。

DECREMENT (*v*)

等同于 INCREMENT (*v*)，将 `allowedValueRange` 步骤作为 `-step`。

DECREMENT_BOUNDED (*v*)

等同于 INCREMENT_BOUNDED (*v*)，将 `allowedValueRange` 步骤作为 `-step`。

DECREMENT_WRAP (v)

等同于 INCREMENT_WRAP(v), 将 allowedValueRange 步骤作为 $-step$ 。

INCREMENT (v)

变量 v 成为 v 加 allowedValueRange 步骤的值, 即 $v = v + step$ 。等同于 DECREMENT (v), 将 allowedValueRange 步骤作为 $-step$ 。 v 必须有一个数字数据类型, 而且必须有一个 allowedValueRange 定义。

INCREMENT_BOUNDED (v)

变量 v 成为 v 加 allowedValueRange 步骤的值, 即 $v = v + step$ 。

如果步骤大于 0 而且 $v + step$ 将大于 allowedValueRange 最大值, 那么 v 成为最大值。

如果步骤小于 0 而且 $v + step$ 将小于 allowedValueRange 最小值, 那么 v 成为最小值。

等同于 DECREMENT_BOUNDED (v), 将 allowedValueRange 步骤作为 $-step$ 。 v 必须有一个数字数据类型, 而且必须有一个 allowedValueRange 定义。

INCREMENT_WRAP (v, c)

变量 v 成为 v 加 allowedValueRange 步骤的值, 即 $v = v + step$ 。

如果步骤大于 0, 而且 $v + step$ 将大于 allowedValueRange 最大值, 那么 v 将成为最小值加步骤减 1, 即 $v = minimum + step - 1$; 如果步骤为 1, 则简化为 $v = minimum$ 。

如果步骤小于 0, 而且 $v + step$ 将小于 allowedValueRange 最小值, 那么 v 将成为最大值加步骤加 1, 即 $v = maximum + step + 1$; 如果步骤为 -1, 则简化为 $v = maximum$ 。

等同于 `DECREMENT_WRAP (v)`，将 `allowedValueRange` 步骤作为 `-step`。`v` 必须有一个数字数据类型，而且必须有一个 `allowedValueRange` 定义。

`NEXT_STRING_BOUNDED (v)`

变量 `v` 将成为 `v` 的当前值之后的下一个 `allowedValue`。如果 `v` 已经是最后一个 `allowedValue`，那么 `v` 没有变化。`v` 必须是一个字符串数据类型，而且必须有一个 `allowedValueList` 定义。

`NEXT_STRING_WRAP (v)`

变量 `v` 将成为 `v` 的当前值之后的下一个 `allowedValue`。如果 `v` 已经是最后一个 `allowedValue`，那么 `v` 将成为第一个 `allowedValue`。`v` 必须是一个字符串数据类型，而且必须有一个 `allowedValueList` 定义。

`PREV_STRING_BOUNDED (v)`

变量 `v` 将成为 `v` 的当前值之前的上一个 `allowedValue`。如果 `v` 已经是第一个 `allowedValue`，那么 `v` 没有变化。`v` 必须是一个字符串数据类型，而且必须有一个 `allowedValueList` 定义。

`PREV_STRING_WRAP (v)`

变量 `v` 将成为 `v` 的当前值之前的上一个 `allowedValue`。如果 `v` 已经是第一个 `allowedValue`，那么 `v` 将成为最后一个 `allowedValue`。`v` 必须是一个字符串数据类型，而且必须有一个 `allowedValueList` 定义。

`SET (v, c)`

变量 `v` 成为常量 `c` 的值，即 `v = c`。`v` 和 `c` 必须是相同的数据类型。

`TOGGLE (v)`

变量 `v` 将成为 `v` 的 `boolean negation` 值，即 `v = NOT v`。`v` 必须是布尔 (`boolean`) 值。

2.9 描述：取得描述

如上面所说明的，在控制点发现了一个设备之后，控制点仍然对设备知之甚少。为了让控制点更多地了解设备和它的功能，控制点必须从发现消息中得到设备描述的 URL，通过 URL 取得 UPnP 设备描述。然后，控制点必须使用设备描述中提供的 URL 取得一个或多个服务描述。这是一个基于 HTTP 的简单流程，使用总体 UPnP 协议栈的下列子集。（总体 UPnP 协议栈在本文开始处列出。）

UPnP 厂商[紫色]
UPnP 论坛[红色]
UPnP 设备架构[绿色]
HTTP[黑色]
TCP[黑色]
IP[黑色]

在最高一层，描述消息包含具体的厂商信息，如设备类型、服务类型和要求的服 务。在协议栈接下来的一层，通过来自 UPnP 论坛工作委员会的信息（如型号名称、型号和具体 URL）对厂商内容进行了补充。来自以上各层的消息存储在 UPnP 特定协议中，并在本文中进行了定义。依次，上述消息均通过 IP 上的 TCP 上的 HTTP 来提供。为便于参考，上面[方括弧]中的颜色显示了何种协议定义下面所列描述消息中的具体标头和消息体元素。

使用这一协议栈，取得 UPnP 设备描述就十分简单：控制点向发现消息中的 URL 发出 HTTP GET 请求，设备在 HTTP 响应的消息体中返回其描述。同样，在取得 UPnP 服务描述时，控制点向设备描述中的 URL 发出 HTTP GET 请求，设备在 HTTP 响应的消息体中返回其描述。响应和请求的标头和消息体将在下面详细说明。

首先，一个控制点必须以下列格式发送一个带有方法 GET 的请求。用斜体表示的值为实际值的占位符。

```
GET path to description HTTP/1.1
```

```
HOST:host for description:port for description
```

```
ACCEPT-LANGUAGE:language preferred by control point
```

（无面向取得描述的请求的消息体，但请注意，消息与最后一个 HTTP 标头之间必须空一行。）

以下列出了上述列表中出现的命令行和标头的详细信息。除特别说明外，所有标头值要区分大小写。

命令行

GET

方法由 HTTP 定义。

描述的路径

设备描述 URL（发现消息中的 LOCATION 标头）或服务描述 URL（设备描述中的 SCPDURL 元素）的路径部分。单一、相对 URL。

HTTP/1.1

HTTP 版本。

标头

HOST

要求。设备描述 URL（发现消息中的 LOCATION 标头）或服务描述 URL（设备描述中的 SCPDURL 元素）的域名或 IP 地址及可选端口部分。如果该端口为空或未给出，则假定为端口 80。

ACCEPT-LANGUAGE

对于取得设备的描述。用于描述的优选语言。如果该语言中没有描述可用，则设备可以返回使用缺省语言的描述。RFC1766 语言标签。

在控制点发送请求之后，设备采取第二个步骤，用描述的副本做出响应。其中包括预期的传输时间，设备必须在 30 秒内做出响应。如果在这段时间内没有响应，控制点应该再次发送请求。设备必须采用以下格式发送响应：用斜体表示的值为实际值的占位符。

```
HTTP/1.1 200 OK
CONTENT-LANGUAGE:language used in description
CONTENT-LENGTH:Bytes in body
CONTENT-TYPE:text/xml
DATE:when responded
```

正如上面的详细说明所指出的，这一响应的消息体是 UPnP 设备或服务描述。

以上标头清单所含信息如下：除特别说明外，所有标头值要区分大小写。

标头

CONTENT-LANGUAGE

仅当请求包括 ACCEPT-LANGUAGE 标头时要求。描述的语言。RFC1766 语言标签。

CONTENT-LENGTH

要求。消息体长度以字节来表示，为整数。

CONTENT-TYPE

要求。必须是 text/xml。

DATE

推荐使用。响应生成时间。RFC1123 日期。

SERVER

(描述消息不要求 SERVER 标头。)

2.10 描述参考

FXPP

灵活的 XML 处理框架。规定未知的 XML 元素及其子元素必须被忽略。
IETF 草案。

ISO 8601

ISO (国际标准化组织)。用日期和时间来表示, 1988-06-15。公布在:
<<http://www.iso.ch/markete/8601.pdf>>。

RFC1123

包括日期格式, 如 HTTP DATE 标头。IETF RFC。

RFC1766

语言标签的格式, 如 HTTP ACCEPT-LANGUAGE 标头。IETF RFC。

RFC2387

代表内容类型的格式, 如用于图标的 mime-type 元素。IETF RFC。

UPC

通用产品代码。12 位全数字代码, 用于确定销售包装。由统一编码委员会
进 行 管 理。
<http://www.uc-council.org/main/ID_Numbers_and_Bar_Codes.html>。

XML

扩展标记语言。W3C 推荐。

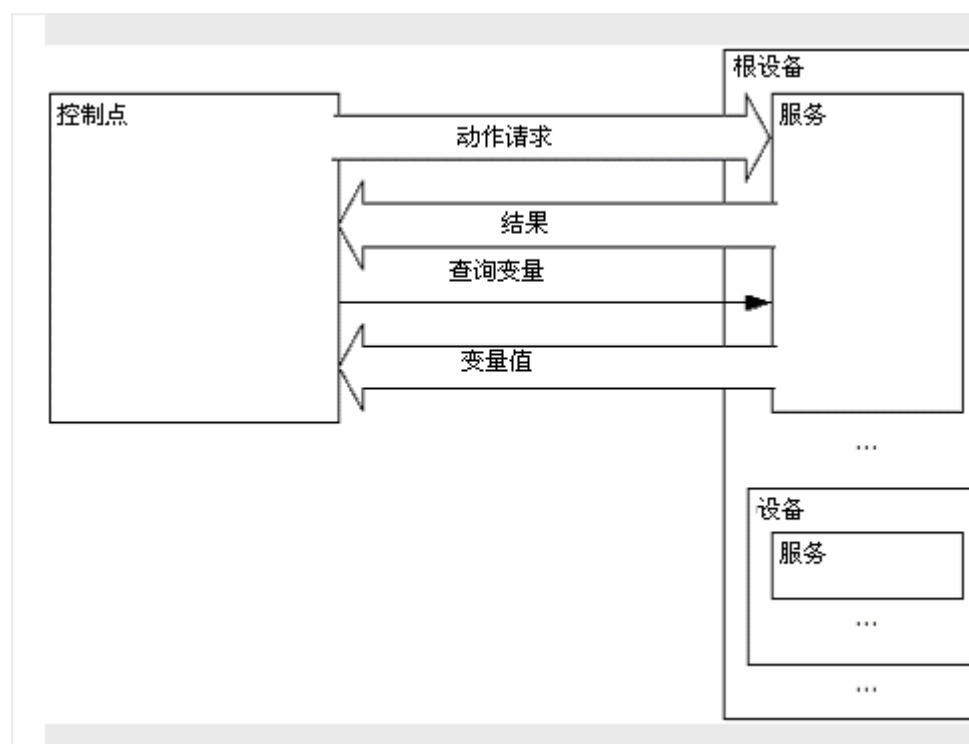
XML 模式 (部分 1: 结构, 部分 2: 数据类型)。

语法定义 UPnP 模板语言。使用 XML 进行定义。W3C 工作草案。部分 1:
结构。部分 2: 数据类型。

3. 控制:

控制是 UPnP 网络的第 3 步。首先是“寻址”（第 0 步），设备会获得一个网络地址；之后是“发现”（第 1 步），控制点会发现相关设备；再后是“描述”（第 2 步），控制点会了解设备能力。控制独立于事件（第 4 步），通过事件控制点可了解设备的状态变化。通过控制，控制点向设备发出动作，并轮询相关状态值。控制和事件是展示（第 5 步）的补充，通过展示控制点会显示设备提供的用户界面。

控制是 UPnP 网络的第 3 步。在接收设备和服务描述之后，控制点可以向这些服务发出动作，同时控制点也可以轮询服务的状态变量值。发出动作实质上是一种远程过程调用；控制点将动作发送到设备服务，在动作完成（或失败）后，服务返回相应的结果或错误。状态变量值轮询是这种场景下的特例，动作及其结果都是预定义的。



为了控制一个设备，控制点向设备服务发出一个动作。这一般由控制点向服务的控制 URL 地址（在设备描述的服务元素 controlURL 子元素部分提供）发送一个

适当的控制消息。而服务则会对此动作做出响应，返回相关结果或错误。动作的效果可以通过改变描述服务运行时状态的变量进行建模。在这些状态变量改变时，事件将被发布到所有相关的控制点。该部分介绍了控制消息的协议栈和格式。“事件”部分介绍了事件发布。

控制点可能会轮询服务的状态变量值以获得状态变量的当前值。与发出一个动作的过程相似，控制点向服务的控制 URL 发送一个适当的查询消息。而服务则返回相应的变量值；每个服务必须保持状态表的一致性，以便控制点能够轮询并接收到有意义的值。本部分还介绍了这些查询消息的格式。“事件”部分介绍了变量值的自动通知。

只要设备的发现宣告没有过期，控制点就会假定该设备及其服务可用。如果设备取消其宣告，则控制点必将认为该设备及其服务不再可用。

UPnP 定义了发出动作和轮询变量值的方法，但 UPnP 不会规定或约束在控制点上运行应用的 API 设计；操作系统厂商可以创建满足其客户需求的 API。

本章节其余部分详细介绍了如何对控制和查询消息进行格式编排并发送到设备。

3.1 控制：协议

要发出动作和轮询变量值，控制点（和设备）会使用以下总体 UPnP 协议栈的子集。（总体 UPnP 协议栈在本文开始处列出。）

UPnP 厂商[紫色]
UPnP 论坛[红色]
UPnP 设备架构[绿色]
SOAPI[蓝色]

HTTP[黑色]
TCP[黑色]
IP[黑色]

在最高一层，控制消息包括特定的厂商信息，如变量值。在协议栈接下来的一层，通过来自 UPNP 论坛工作委员会的信息（如动作名称、变量名称、变量名称）对厂商内容进行了补充。来自以上各层的消息存储在 UPNP 特定协议中，并在本文中进行了定义。采用简单对象访问协议（SOAP）标头和消息体元素依次对上述消息进行格式编排，并通过 HTTP、TCP/IP 进行发送。为便于参考，上文[方括弧]中的颜色显示出哪个协议定义下列订阅消息中的具体标头元素。

3.2 控制：动作

控制点会向设备的服务发出动作，并接收结果或错误返回。该动作、结果和错误密封在 SOAP 中，通过 HTTP 请求发送，并通过 HTTP 响应接收。

3.2.1 控制：动作：调用

简单对象访问协议（SOAP）定义使用 XML 和 HTTP 的远程过程调用。UPnP 使用 SOAP 来向设备提供控制消息，并将结果或错误返回控制点。

SOAP 定义额外的 HTTP 标头，以确保这些不会与其它 HTTP 扩展混淆，SOAP 遵循 HTTP 扩展框架，在 MAN 标识中指定 SOAP 独特 URI，并在 HTTP 方法中采用 M-作为前缀。在这种情况下，此方法为 M-POST。使用 M-POST 要求 HTTP 服务器发现并了解 SOAP 独特 URI 和 SOAP 特定标头。

为了向防火墙和代理提供更大的管理灵活性，SOAP 规定首先必须在不带 MAN 标头或 M-前缀的情况下尝试请求。如果请求被拒绝并返回“405 Method Not Allowed”信息，则必须使用 MAN 标头和 M-前缀发送第二个请求。如果请求拒

绝并返回“501 Not Implemented”或“510 Not Extended”信息，则请求失败。
(其它 HTTP 请求应根据 HTTP 规范进行处理。)

下面是一个使用 POST 方法发送的控制消息列表（没有 MAN 标头），随后是标头和消息体的解释。在这之后是一个采用 M-POST 方法和 MAN 标头发送的控制消息列表。

为了向设备的服务发出一个动作，控制点必须采用以下格式的 POST 方法发送一个请求。用斜体表示的值为实际值的占位符。

```
POST path of control URL HTTP/1.1
HOST: host of control URL:port of control URL
CONTENT-LENGTH: bytes in body
CONTENT-TYPE: text/xml; charset="utf-8"
SOAPACTION: "urn:schemas-upnp-org:service:serviceType:v#actionName"

<s:Envelope
  xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <s:Body>
    <u:actionName xmlns:u="urn:schemas-upnp-org:service:serviceType:v">
      <argumentName>in arg value</argumentName>
      other in args and their values go here, if any
    </u:actionName>
  </s:Body>
</s:Envelope>
```

以下列出了上述列表中出现的命令行、标头和消息体元素的详细信息。所有标头值和元素名称均区分大小写；除特别注明外，值不区分大小写。除特别注明外，元素的顺序无关紧要。除特别注明外，所要求的元素必须出现一次（无重复），推荐或可选元素最多只能出现一次。

命令行

POST

方法由 HTTP 定义。

路径控制 URL。

控制此服务的 URL 路径组件(设备描述服务元素的 controlURL 子元素)。

单一、相对 URL。

HTTP/1.1

HTTP 版本。

标头

HOST

要求。控制此服务的域名或 IP 地址、以及 URL 可选端口组件(设备描述服务元素的 controlURL 子元素)。如果该端口为空或未给出,则假定为端口 80。

ACCEPT-LANGUAGE

(控制消息中未使用 ACCEPT-LANGUAGE 标头。)

CONTENT-LENGTH

要求。消息体长度以字节来表示,为整数。

CONTENT-TYPE

要求。必须是 text/xml。应包括使用的字符编码,如 utf-8。

MAN

(采用方法 POST 不要求 MAN 标头。)

SOAPACTION

SOAP 规定使用的标头。必须是要调用的服务类型、散列符号和动作名称,均用双引号括起来。如果用于采用 M-POST 方法的请求,标头名称必须符合 MAN 标头中定义的 HTTP 名字空间。单一 URI。

消息体:

信包

SOAP 规定使用的元素。Xmlns 名字空间属性必须为“<http://schemas.xmlsoap.org/soap/envelope/>”。必须包括带有值为“<http://schemas.xmlsoap.org/soap/encoding/>”的 `encodingStyle` 属性。

包含以下子元素：

消息体

SOAP 规定使用的元素。应符合 SOAP 名字空间。包含以下子元素：

actionName

要求。元素名称即要调用的动作名称。xmlns 名字空间属性必须为双引号中包含的服务类型。必须为 `消息体` 的第一个子元素。包含以下有序子元素：

argumentName

如果且仅当动作带有输入变量时要求。传送给动作的值。输入变量中的每个动作重复一次。（元素名称不符合名字空间；元素嵌套上下文足够。）单一数据类型由 UPnP 服务描述定义。

为了获得未来可伸缩性，在处理上述 XML 时要遵守灵活 XML 处理框架(FXPP)，设备和控制点必须忽略：**(a)** 任何未知元素及其子元素或内容，以及**(b)**任何未知属性及其值。

如果一个带有 POST 的请求被拒绝并返回“405 Method Not Allowed”信息，则控制点必须采用以下格式的 M-POST 和 MAN 方法发送第二个请求。用斜体表示的值为实际值的占位符。

M-POST *path of control URL* HTTP/1.1

HOST: *host of control URL:port of control URL*

CONTENT-LENGTH: *bytes in body*

CONTENT-TYPE: text/xml; charset="utf-8"

MAN: "http://schemas.xmlsoap.org/soap/envelope/"; ns=01

01-SOAPACTION: "urn:schemas-upnp-org:[service](#):*serviceType*:*v#actionName*"

（带有 M-POST 方法的请求消息消息体与带有 POST 方法的请求消息体相同。详见如上说明。）

命令行

M-POST

方法由 HTTP 扩展框架定义。

控制 URL 的路径。

控制此服务的 URL 路径组件(设备描述服务元素的 controlURL 子元素)。

单一、相对 URL。

HTTP/1.1

HTTP 版本。

标头

HOST

要求。控制此服务的域名或 IP 地址、以及 URL 可选端口组件（设备描述服务元素的 controlURL 子元素）。如果该端口为空或未给出，则假定为端口 80。

ACCEPT-LANGUAGE

（控制消息中未使用 ACCEPT-LANGUAGE 标头。）

CONTENT-LENGTH

要求。消息体长度以字节来表示，为整数。

CONTENT-TYPE

要求。必须是 text/xml。应包括使用的字符编码，如 utf-8。

MAN

要求。必须为 “http://schemas.xmlsoap.org/soap/envelope/”。ns 指令定义用于其它 SOAP 标头的名字空间（如 01）（如 [SOAPACTION](#)）。

SOAPACTION

SOAP 规定使用的标头。必须是要调用的服务类型、散列符号和动作名称，均用双引号括起来。如果用于采用 M-POST 方法的请求，标头名称必须符合 MAN 标头中定义的 HTTP 名字空间。单一 URI。

3.2.2 控制：动作：响应

该服务必须完成动作调用，并在 30 秒内响应，包括预计的传送时间。应对需要用较长时间的动作进行定义，以便在动作完成后尽早返回并发送一个事件。如果服务在此时间内未能响应，则控制点的操作应视具体应用而定。服务必须采用以下格式发送响应：用斜体表示的值为实际值的占位符。

```
HTTP/1.1 200 OK
CONTENT-LENGTH: bytes in body
CONTENT-TYPE: text/xml; charset="utf-8"
DATE: when response was generated
EXT:
SERVER: OS/version UPnP/1.0 product/version
```

```
<s:Envelope
  xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <s:Body>
    <u:actionNameResponse xmlns:u="urn:schemas-upnp-org:service:serviceType:v">
      <argumentName>out arg value</argumentName>
      other out args and their values go here, if any
    </u:actionNameResponse>
  </s:Body>
</s:Envelope>
```

以下列出了上述列表中出现的命令行、标头和消息体元素的详细信息。所有标头值和元素名称均区分大小写；除特别注明外，值不区分大小写。除特别注明外，元素的顺序无关紧要。除特别注明外，所要求的元素必须出现一次（无重复），推荐或可选元素最多只能出现一次。

响应行

HTTP/1.1

HTTP 版本。

200 OK

HTTP 成功代码。

标头

CONTENT-LANGUAGE

（控制消息中未使用 **ACCEPT-LANGUAGE** 标头。）

CONTENT-LENGTH

要求。消息体长度以字节来表示，为整数。

CONTENT-TYPE

要求。必须是 **text/xml**。应包括使用的字符编码，如 **utf-8**。

DATE

推荐使用。响应生成时间。**RFC1123** 日期。

EXT

要求。确定 **MAN** 标头已被理解。（只限标头，无值。）

SERVER

要求。一系列操作系统名称、操作系统版本、**UPnP/1.0**、产品名称以及产品版本信息。字符串。

消息体

信包

SOAP 规定使用的元素。xmlns 名字空间属性必须为“<http://schemas.xmlsoap.org/soap/envelope/>”。必须包括带有值为“<http://schemas.xmlsoap.org/soap/encoding/>”的[encodingStyle](#)属性。

包含以下子元素：

消息体

SOAP 规定使用的元素。应符合 SOAP 名字空间。包含以下子元素：

actionNameResponse

要求。元素名称即前置于[响应](#)的动作名称。xmlns 名字空间属性必须为双引号中包含的服务类型。必须为[消息体](#)的第一个子元素。包含以下子元素：

argumentName

如果且仅当动作带有输出变量时要求。值从动作中返回。输出变量中的每个动作重复一次。如果动作带有标记为 `retval` 的变量，则此变量必须为第一个元素。（元素名称不符合名字空间；元素嵌套上下文足够。）单一数据类型由 UPnP 服务描述定义。

为了获得未来可伸缩性，在处理上述 XML 时要遵守灵活 XML 处理框架(FXPP)，设备和控制点必须忽略：**(a)** 任何未知元素及其子元素或内容，以及**(b)**任何未知属性及其值。

如果服务在调用控制点发送的动作时出现错误，则服务必须在 30 秒内发送响应，包括预计的传送时间。不能使用输出变量来传达错误信息；输出变量只能用于返回数据；错误响应必须采用以下格式发送。用斜体表示的值为实际值的占位符。

```
HTTP/1.1 500 Internal Server Error
CONTENT-LENGTH: bytes in body
CONTENT-TYPE: text/xml; charset="utf-8"
DATE: when response was generated
EXT:
SERVER: OS/version UPnP/1.0 product/version
```

```
<s:Envelope
```

```
  xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
```

```
  s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
```

```
    <s:Body>
```

```
      <s:Fault>
```

```
        <faultcode>s:Client</faultcode>
```

```
<faultstring>UPnPError</faultstring>
<detail>
  <UPnPError xmlns="urn:schemas-upnp-org:control-1-0">
    <errorCode>error code</errorCode>
    <errorDescription>error string</errorDescription>
  </UPnPError>
</detail>
</s:Fault>
</s:Body>
</s:Envelope>
```

以下列出了上述列表中出现的命令行、标头和消息体元素的详细信息。所有标头值和元素名称均区分大小写；除特别注明外，值不区分大小写。除特别注明外，元素的顺序无关紧要。除特别注明外，所要求的元素必须出现一次（无重复），推荐或可选元素最多只能出现一次。

响应行

HTTP/1.1

HTTP 版本。

500 内部服务器错误

HTTP 错误代码。

标头

CONTENT-LANGUAGE

（控制消息中未使用 ACCEPT-LANGUAGE 标头。）

CONTENT-LENGTH

要求。消息体长度以字节来表示，为整数。

CONTENT-TYPE

要求。必须是 text/xml。应包括使用的字符编码，如 utf-8。

DATE

推荐使用。响应生成时间。RFC1123 日期。

EXT

要求。确定 MAN 标头已被理解。（只限标头，无值。）

SERVER

要求。一系列操作系统名称、操作系统版本、[UPnP/1.0](#)、产品名称以及产品版本信息。字符串。

消息体

信包

SOAP 规定使用的元素。xmlns 名字空间属性必须为“http://schemas.xmlsoap.org/soap/envelope/”。必须包括带有值为“http://schemas.xmlsoap.org/soap/encoding/”的[encodingStyle](#)属性。

包含以下子元素：

消息体

SOAP 规定使用的元素。应符合 SOAP 名字空间。包含以下子元素：

错误

SOAP 规定使用的元素。调用动作时遇到的错误。应符合 SOAP 名字空间。包含以下子元素：

faultcode

SOAP 规定使用的元素。值必须符合 SOAP 名字空间。值必须为 [Client](#)。

faultstring

SOAP 规定使用的元素。值必须为 [UPnPError](#)。

详细信息

SOAP 规定使用的元素。

UPnPError

UPnP 规定使用的元素。

errorCode

UPnP 规定使用的元素。代码识别遇到过哪些错误。
立即查看下表的值。整数。

errorDescription

UPnP 规定使用的元素。简短描述。立即查看下表的值。字符串。推荐 < 256 字符。

下表概括定义了 errorCode 和 errorDescription 元素的错误类型和相应值。

errorCode	errorDescription	描述
401	Invalid Action	这一服务中该名称没有动作。
402	Invalid Args	可能如下所示: not enough in args、too many in args、no in arg by that name、one or more in args 均为错误数据类型。
403	Out of Sync	不同步
501	Action Failed	可能在当前服务状态下返回, 以避免调用此动作。
600-699	TBD	一般动作错误。由 UPnP 论坛技术委员会定义。
700-799	TBD	面向标准动作的特定动作错误。由 UPnP 论坛工作委员会定义。
800-899	TBD	面向非标准动作的特定动作错误。由 UPnP 厂商定义。

为了获得未来可伸缩性, 在处理上述 XML 时要遵守灵活 XML 处理框架(FXPP), 设备和控制点必须忽略: (a) 任何未知元素及其子元素或内容, 以及(b)任何未知属性及其值。

3.3 控制: 查询变量

除了向设备的服务发出动作, 控制点还可以对服务进行轮询, 以通过发送查询消息获得状态变量值。查询消息只能查询一个状态变量, 必须发送多个查询消息以查询多个状态变量。

此查询消息与该服务的事件（如果有）相分离。如果变量适度，则变量值查询通常会生成比通过事件触发收到的值更新的值。Eventing 部分描述了事件适度。

3.3.1 控制：查询：调用

要查询状态变量值，控制点必须采用以下格式发送请求。用斜体表示的值为实际值的占位符。

```
POST path of control URL HTTP/1.1
HOST: host of control URL:port of control URL
CONTENT-LENGTH: bytes in body
CONTENT-TYPE: text/xml; charset="utf-8"
SOAPACTION: "urn:schemas-upnp-org:control-1-0#QueryStateVariable"
```

```
<s:Envelope
  xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <s:Body>
    <u:QueryStateVariable xmlns:u="urn:schemas-upnp-org:control-1-0">
      <u:varName>variableName</u:varName>
    </u:QueryStateVariable>
  </s:Body>
</s:Envelope>
```

以下列出了上述列表中出现的命令行、标头和消息体元素的详细信息。所有标头值和元素名称均区分大小写；除特别注明外，值不区分大小写。除特别注明外，元素的顺序无关紧要。除特别注明外，所要求的元素必须出现一次（无重复），推荐或可选元素最多只能出现一次。

命令行

POST

方法由 HTTP 定义。

控制 URL 的路径。

控制此服务的 URL 路径组件(设备描述服务元素的 controlURL 子元素)。

单一、相对 URL。

HTTP/1.1

HTTP 版本。

标头

HOST

要求。控制此服务的域名或 IP 地址、以及 URL 可选端口组件(设备描述服务元素的 controlURL 子元素)。如果该端口为空或未给出,则假定为端口 80。

ACCEPT-LANGUAGE

(控制消息中未使用 ACCEPT-LANGUAGE 标头。)

CONTENT-LENGTH

要求。消息体长度以字节来表示,为整数。

CONTENT-TYPE

要求。必须是 text/xml。应包括使用的字符编码,如 utf-8。

MAN

(采用方法 POST 不要求 MAN 标头。)

SOAPACTION

SOAP 规定使用的标头。必须为“urn : [schemas-upnp-org:control-1-0#QueryStateVariable](#)”。如果用于采用 M-POST 方法的请求,标头名称必须符合 MAN 标头中定义的 HTTP 名字空间。单一 URI。

消息体

信包

SOAP 规定使用的元素。xmlns 名字空间属性必须为“http://schemas.xmlsoap.org/soap/envelope/”。必须包括带有值为“http://schemas.xmlsoap.org/soap/encoding/”的[encodingStyle](#)属性。

包含以下子元素：

消息体

SOAP 规定使用的元素。应符合 SOAP 名字空间。包含以下子元素：

QueryStateVariable

UPnP 规定使用的元素。动作名称。xmlns 名字空间属性必须为“urn: [schemas-upnp-org:control-1-0](#)”。必须为消息体的第一个子元素。包含以下顺序排列的子元素：

varName

UPnP 规定使用的元素。变量名称。必须符合 [QueryStateVariable](#) 名字空间。值是要查询的状态变量的名称。字符串。

为了获得未来可伸缩性,在处理上述 XML 时要遵守灵活 XML 处理框架(FXPP),设备和控制点必须忽略: (a) 任何未知元素及其子元素或内容, 以及(b)任何未知属性及其值。

如果一个带有 POST 的请求被拒绝并返回“405 Method Not Allowed”信息,则控制点必须采用上述 M-POST 和 MAN 方法发送第二个请求。

3.3.2 控制：查询：响应

要响应状态变量值的查询,服务必须在 30 秒内响应,包括预计的传送时间。如果服务在此时间内未能响应,则控制点的操作应视具体应用而定。服务必须采用以下格式发送响应: 用*斜体*表示的值为实际值的占位符。

HTTP/1.1 200 OK

CONTENT-LENGTH: *bytes in body*

CONTENT-TYPE: text/xml; charset="utf-8"
DATE: *when response was generated*
EXT:
SERVER: *OS/version UPnP/1.0 product/version*

```
<s:Envelope  
  xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"  
  s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">  
  <s:Body>  
    <u:QueryStateVariableResponse xmlns:u="urn:schemas-upnp-org:control-1-0">  
      <return>variable value</return>  
    </u:QueryStateVariableResponse>  
  </s:Body>  
</s:Envelope>
```

以下列出了上述列表中出现的命令行、标头和消息体元素的详细信息。所有标头值和元素名称均区分大小写；除特别注明外，值不区分大小写。除特别注明外，元素的顺序无关紧要。除特别注明外，所要求的元素必须出现一次（无重复），推荐或可选元素最多只能出现一次。

响应行

HTTP/1.1

HTTP 版本。

200 OK

HTTP 成功代码。

标头

CONTENT-LANGUAGE

（控制消息中未使用 ACCEPT-LANGUAGE 标头。）

CONTENT-LENGTH

要求。消息体长度以字节来表示，为整数。

CONTENT-TYPE

要求。必须是 **text/xml**。应包括使用的字符编码，如 **utf-8**。

DATE

推荐使用。响应生成时间。**RFC1123** 日期。

EXT

要求。确定 **MAN** 标头已被理解。（只限标头，无值。）

SERVER

要求。一系列操作系统名称、操作系统版本、**UPnP/1.0**、产品名称以及产品版本信息。字符串。

消息体

信包

SOAP 规定使用的元素。xmlns 名字空间属性必须为“<http://schemas.xmlsoap.org/soap/envelope/>”。必须包括带有值为“<http://schemas.xmlsoap.org/soap/encoding/>”的 [encodingStyle](#) 属性。

包含以下子元素：

消息体

SOAP 规定使用的元素。应符合 **SOAP** 名字空间。包含以下子元素：

QueryStateVariableResponse

所要求的元素由 **UPnP** 和 **SOAP** 定义。xmlns 名字空间属性必须为“urn: [schemas-upnp-org:control-1-0](#)”。必须为 [消息体](#) 的第一个子元素。包含以下子元素：

返回

UPnP 规定使用的元素。（元素名称不符合名字空间；元素嵌套上下文足够。）值为当前状态变量值，应要求在 [varName](#) 元素中列出。

为了获得未来可伸缩性,在处理上述 XML 时要遵守灵活 XML 处理框架(FXPP),设备和控制点必须忽略:(a) 任何未知元素及其子元素或内容,以及(b) 任何未知属性及其值。

如果服务不能为此变量提供一个值,则该服务必须在 30 秒内发送一个响应,包括预计的传送时间。响应必须采用以下格式:用斜体表示的值为实际值的占位符。

```
HTTP/1.1 500 Internal Server Error
CONTENT-LENGTH: bytes in body
CONTENT-TYPE: text/xml; charset="utf-8"
DATE: when response was generated
EXT:
SERVER: OS/version UPnP/1.0 product/version
```

```
<s:Envelope
  xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <s:Body>
    <s:Fault>
      <faultcode>s:Client</faultcode>
      <faultstring>UPnPError</faultstring>
      <detail>
        <UPnPError xmlns="urn:schemas-upnp-org:control-1-0">
          <errorCode>error code</errorCode>
          <errorDescription>error string</errorDescription>
        </UPnPError>
      </detail>
    </s:Fault>
  </s:Body>
</s:Envelope>
```

以下列出了上述列表中出现的命令行、标头和消息体元素的详细信息。所有标头值和元素名称均区分大小写;除特别注明外,值不区分大小写。除特别注明外,元素的顺序无关紧要。除特别注明外,所要求的元素必须出现一次(无重复),推荐或可选元素最多只能出现一次。

响应行

HTTP/1.1

HTTP 版本。

500 内部服务器错误

HTTP 错误代码。

标头

CONTENT-LANGUAGE

（控制消息中未使用 `ACCEPT-LANGUAGE` 标头。）

CONTENT-LENGTH

要求。消息体长度以字节来表示，为整数。

CONTENT-TYPE

要求。必须是 `text/xml`。应包括使用的字符编码，如 `utf-8`。

DATE

推荐使用。响应生成时间。`RFC1123` 日期。

EXT

要求。确定 `MAN` 标头已被理解。（只限标头，无值。）

SERVER

要求。一系列操作系统名称、操作系统版本、[UPnP/1.0](#)、产品名称以及产品版本信息。字符串。

消息体

信包

SOAP 规定使用的元素。xmlns 名字空间属性必须为“`http://schemas.xmlsoap.org/soap/envelope/`”。必须包括带有值为“`http://schemas.xmlsoap.org/soap/encoding/`”的[encodingStyle](#)属性。

包含以下子元素：

消息体

SOAP 规定使用的元素。应符合 SOAP 名字空间。包含以下子元素：
错误

SOAP 规定使用的元素。为何该服务不会为变量返回一个值。应符合 SOAP 名字空间。包含以下子元素：

faultcode

SOAP 规定使用的元素。应符合 SOAP 名字空间。值必须为 [Client](#)。

faultstring

SOAP 规定使用的元素。描述 **errorCode** 的一般 UPnP 字符串。立即查看下表的值。

detail

SOAP 规定使用的元素。包含以下子元素：

UPnPError

UPnP 规定使用的元素。包含以下子元素：

errorCode

UPnP 规定使用的元素。代码识别遇到过哪些错误。立即查看下表的值。整数。

errorDescription

UPnP 规定使用的元素。简短描述。立即查看下表的值。字符串。推荐 < 256 字符。

下表概括定义了 **errorCode** 和 **errorDescription** 元素的错误类型和相应值。

errorCode	errorDescription	描述
404	Invalid Var	这一服务中该名称没有动作。
600-624	TBD	一般动作错误。由 UPnP 论坛技术委员会定义。
625-649	TBD	留作未来使用。
650-674	TBD	面向标准动作的特定动作错误。由 UPnP 论坛工作委员会定义。

675-699	TBD	面向非标准动作的特定动作错误。由 UPnP 厂商定义。
---------	-----	-----------------------------

为了获得未来可伸缩性,在处理上述 XML 时要遵守灵活 XML 处理框架(FXPP),设备和控制点必须忽略:(a) 任何未知元素及其子元素或内容,以及(b)任何未知属性及其值。

3.4 控制参考

FXPP

灵活的 XML 处理框架。规定未知的 XML 元素及其子元素必须被忽略。
IETF 草案。

HTTP 扩展框架

描述针对 HTTP 的通用扩展机制。W3C RFC。

RFC1123

包括日期格式,如 HTTP DATE 标头。IETF RFC。

SOAP

简单对象访问协议。针对远程过程调用,用 XML 定义的、在 HTTP 之上的协议。IETF 草稿和 W3C 技术报告。

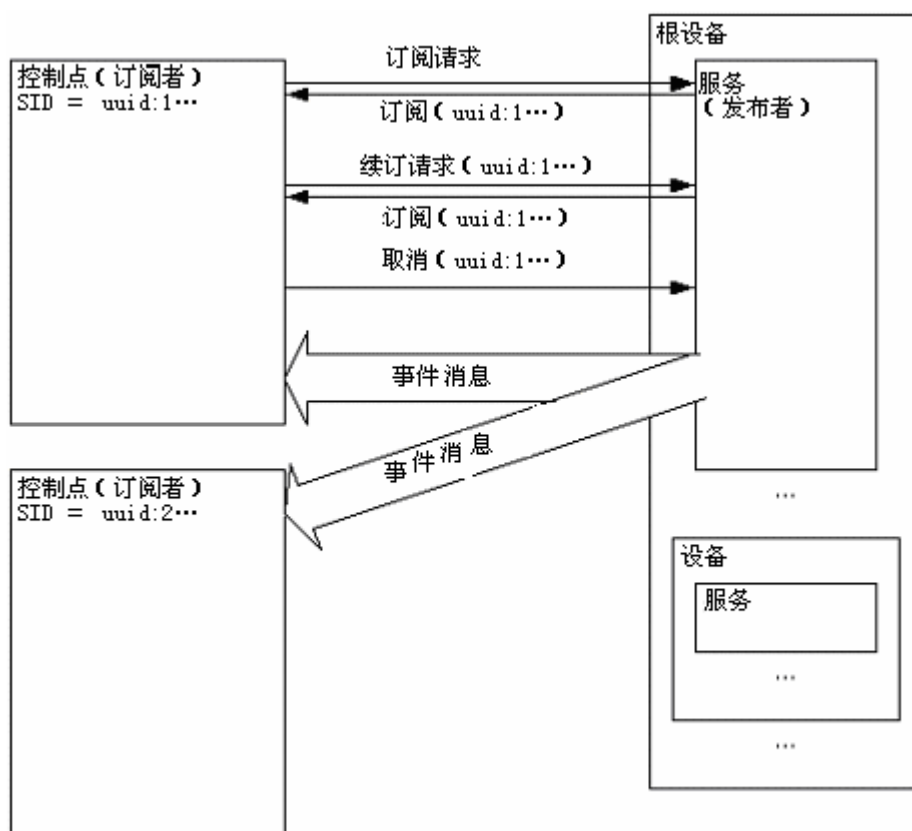
XML

扩展标记语言。W3C 推荐。

4. 事件

事件是 UPnP 网络的第 4 步。首先是“寻址”(第 0 步),设备会获得一个网络地址;之后是“发现”(第 1 步),控制点会发现相关设备;再后是“描述”(第 2 步),控制点会了解设备能力。“事件触发”与“控制”(第 3 步)紧密相连,第 3 步中控制点向设备发送动作。通过“事件触发”,控制点监听设备的状态变化。控制和事件触发是展示(第 5 步)的补充,通过展示控制点会显示设备提供的用户界面。

控制点发现（1）设备和（2）取得该设备及其服务的描述之后，就拥有了事件触发功能的要素。正如描述部分所述，一个即插即用服务描述包括服务响应的动作列表和运行时描述服务状态的变量列表。如果一个或多个状态变量可以被事件触发，服务将会在这些变量发生变化时发布更新，控制点可以订阅以获得此信息。通过这一部分，*发布者*指事件的来源（通常为设备服务），*订阅者*指事件目的地（通常为控制点）。

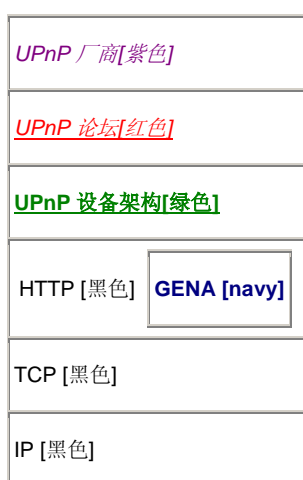


要订阅事件，订阅者可发送一条*订阅消息*。如果订阅者收到此消息，将回复此订阅的持续时间。要保持订阅，订阅者必须在订阅过期之前进行续订。当订阅者不再需要发布者发送的事件时，订阅者应当取消其订阅。下面将详细介绍订阅、续订和取消消息的相关信息。

发布者通过发送 *事件消息* 提醒订阅者状态变量改变。事件消息包含多个状态变量名称和这些变量的当前值，以 **XML** 表示。在订阅者第一次订阅时，需要发送一个专门的 *初始化事件消息*。该事件消息包含所有事件变量的名称和值，并且允许订阅者初始化其服务状态模型。为了支持多个控制点，在动作生效后所有订阅者均会收到通知。由此，将向所有订阅者发送全部事件消息，订阅者会收到所有事件触发变量的事件消息（不只是一部分）。不管状态变量因何种原因发生改变（无论为了响应所要求的动作，还是由于服务模拟的状态发生变化）均会发送事件消息。以下部分详细阐明了事件消息格式的相关信息。

某些状态变量值的改变太快，以致事件触发功能没有意义。另一个方法就是过滤或缓和由于变量值变化而发送的事件消息数量。某些状态变量所包含的值过大而使事件触发功能没有意义；因此，或出于其它原因，服务可以将一个或多个状态变量指定为 *非事件触发*，并且永不向订阅者发送事件消息。为确定此类非事件触发变量的当前值，控制点必须要对服务进行明确轮询。本部分将介绍服务描述如何对可变事件触发进行说明。控制部分介绍了如何对服务的变量值进行轮询。

要发送和接收订阅及事件消息，控制点和 service 可采用以下总体 **UPnP** 协议栈的子集。（总体 **UPnP** 协议栈列于本文开始部分）



最高一层的订阅和事件消息包含诸如订阅和订阅持续时间或具体变量值等特定厂商信息。在协议栈接下来的一层，通过来自 **UPnP** 论坛工作委员会的信息（如

服务标识符或变量名)对厂商内容做了补充。根据本文规定,来自上层的消息驻留在 UPnP 特定协议中。而上述消息通过已采用通用事件通知架构(GENA)方法和标头进行扩展的 HTTP 依次被发送。HTTP 消息通过 TCP/IP 进行发送。为便于参考,上文[方括弧]中的颜色显示出哪个协议定义下列订阅消息中的具体标头元素。

本部分其余部分首先对订阅进行了说明,包括订阅信息、续订信息和取消信息的详细情况。其次,详细阐述了事件消息如何进行格式安排并发送到控制点,以及初始化事件消息。最后,介绍了与事件触发相关的 UPnP 模板语言。

4.1 事件: 订阅

当且仅当在一个或多个状态变量被事件触发时,服务才具有事件触发功能。

如果服务具有事件触发功能,它则可向相关订阅者发布事件消息。发布者保留订阅者列表,向每个订阅者传达以下信息。

唯一的订阅标识符

要求。必须保证标识符在订阅期间的唯一性,无论这一时期有多长;由发布者响应订阅消息而生成。建议全部采用唯一的标识符,以确保唯一性;单一 URI。

事件消息交付 URL

要求。由订阅者在订阅消息中提供;单一 URL。

事件编号

要求。初始化事件消息的编号为 0。必须按顺序对每个后续事件消息编号进行排序;如果订阅者收到带有顺序号的事件编号,则可以验证事件消息是否丢失。必须回置到 1。应为 4 字节(32 位)。一个整数。

订阅持续时间

要求。订阅期满前的时间或持续时间；一个整数或关键字“infinite”（无限）。

发布者应视其合理维护和交付能力，接受尽可能多的订阅。

发布者希望在电源故障时仍可接受订阅。如果问题非常很小且仅限于设备，重复利用存储的订阅可以加快恢复，从而使控制点从整个网络故障中得以恢复。

订阅者列表可通过下面介绍的订阅、续订和取消消息以及本部分后面介绍的事件消息进行更新。

要订阅服务事件，订阅者需向发布者发送一个包含发布者 URL、服务标识符和事件消息交付 URL 的 *订阅消息*。订阅消息还应包括要求的订阅持续时间。面向发布者的 URL 和服务标识符可在描述消息中找到。正如描述部分所述，描述消息包含设备描述。就每个设备而言，设备描述均包含（在其它事物中）一个事件 URL（在 eventSubURL 中）和一个服务标识符（在 serviceId 元素中），分别用来对应面向发布者的 URL 和服务标识符。面向发布者的 URL 必须是唯一的，与设备中的特定服务相对应。

订阅消息是一个接收所有事件消息的请求。未提供在变量基础上订阅事件消息的机制。订阅者将接收来自服务的所有事件消息。在设计服务时，应将这一因素考虑在内。

如果发布者接受订阅消息，它将返回唯一的订阅标识符和订阅持续时间。持续时间选择应与控制点从网络上退出出频度的假设相匹配；如果控制点每几分钟就会从网络中退出，那么持续时间也相应较短，以便发布者迅速终止所有期满订阅者；如果控制点的存在为半永久，那么持续时间也应较长，从而最大限度地减少与续订相关的处理和流量。

接受订阅后，发布者应尽快向订阅者发送第一个或*初始化*事件消息。该消息包含所有事件触发变量的名称和当前值。（每个变量的数据类型和范围在设备描述中进行了说明。描述部分对这一问题进行了详细介绍。）

为了保持订阅，订阅者必须在订阅过期之前通过发送续订消息进行续订。续订消息被发送到与订阅消息相同的 **URL**，但续订消息不包括事件消息的交付 **URL**，而是包括订阅标识符。对续订消息的响应与对订阅消息的响应相同。

如果订阅过期，该订阅标识符将失效，发布者停止向订阅者发送事件消息，并会清除其订阅者列表。如果订阅者试图发送订阅消息之外的任何消息，发布者将会以该订阅标识符失效为由而拒收此消息。

当订阅者不再需要来自特定服务的事件触发时，订阅者应取消其订阅。取消订阅通常可以减少服务、控制点和网络负载。如果一个订阅者突然从网络上退出，它可能无法再发送取消消息。根据规定，除非续订否则订阅最终将会期满。

订阅者应时刻注意来自发布者的发现消息。如果发布者取消了其宣告，订阅者应假定其订阅已被取消。

下面对订阅、续订和取消消息的特定格式的请求、响应和错误进行了介绍。

4.1.1 事件触发：订阅：带有 **NT 和 **CALLBACK** 的订阅**

对于设备中的每项服务，描述消息均包含一个事件触发 **URL**（设备描述中服务元素的 **eventSubURL** 子元素）和 **UPnP** 服务标识符（设备描述中服务元素的 **serviceld** 子元素）。为订阅特定服务的事件，订阅消息将被发送到该服务的事件触发 **URL**。（注意，事件触发 **URL** 可能相对于基本 **URL**。）消息中含该服务的标识符以及事件消息交付 **URL**。订阅消息还可能包含所要求的订阅持续时间。

要订阅服务事件，订阅者必须通过以下格式，发送一个采用 **SUBSCRIBE** 方法并带有 **NT** 和 **CALLBACK** 标头的请求。用斜体表示的值为实际值的占位符。

```
SUBSCRIBE publisher path HTTP/1.1  
HOST: publisher host:publisher port  
CALLBACK: <delivery URL>  
NT: upnp:event  
TIMEOUT: Second-requested subscription duration
```

（采用 **SUBSCRIBE** 方法的请求没有消息体，但请注意，该消息与最后一个 HTTP 标头之间必须空一行。）

以下列出了上述列表中出现的命令行和标头的详细信息。除特别注明外，所有标头值均要区分大小写。

命令行

SUBSCRIBE

方法由 **GENA** 定义。开始订阅或续订。

发布者路径

事件触发 URL（设备描述中**服务**元素的 eventSubURL 子元素）路径组成。单一、相对 URL。

HTTP/1.1

HTTP 版本。

标头

HOST

要求。事件触发 URL（设备描述中**服务**元素的 eventSubURL 子元素）的域名或 IP 地址和可选端口组件。如果该缺少或为空，则假定为端口 80。

CALLBACK

GENA 规定使用的标头。事件消息发往的位置。由 UPnP 厂商规定。如果有多于 1 个 URL, 则当服务发送事件时, 它将依次尝试这些 URL 直至成功为止。一个或多个 URL 用角括号分开。

NT

GENA 规定使用的标头。通知类型。必须采用 [upnp:event 形式](#)。

SID

(订阅不采用 SID 标头。)

TIMEOUT

建议采用。直到订阅期满所要求的持续时间, 秒数或无限时间。由 UPnP 论坛工作委员会推荐。由 UPnP 厂商规定。关键字 “second” (秒) 后为一个整数 (无空格) 或关键字 “infinite” (无限)。

如果有足够的资源维持订阅, 发布者应接受它。为接受订阅, 发布者会分配一个唯一的订阅标识符和持续时间, 并发送一个初始化事件消息 (本部分后面将详细介绍)。要接受订阅请求, 发布者必须在 30 秒 (包括预计的传输时间) 内以下列格式发送一个响应。用 *斜体* 表示的值为实际值的占位符。

HTTP/1.1 200 OK

DATE: *when response was generated*

SERVER: *OS/version UPnP/1.0 product/version*

SID: *uuid:subscription-UUID*

TIMEOUT: *Second-actual subscription duration*

(采用 SUBSCRIBE 方法的请求没有消息体, 但请注意, 该消息与最后一个 HTTP 标头之间必须空一行。)

有关上面列表中标头的详细信息如下。除特别注明外, 所有标头值均要区分大小写。

标头

DATE

建议采用。响应生成时间。RFC1123 日期。

SERVER

要求。一系列操作系统名称、操作系统版本、[UPnP/1.0](#)、产品名称以及产品版本信息。字符串。

SID

GENA 规定使用的标头。订阅标识符。必须是唯一的。必须以 `uuid` 开头：由 UPnP 厂商规定。单一 URI。

TIMEOUT

要求。直到订阅期满的实际持续时间，秒数或无限时间。由 UPnP 论坛工作委员会推荐。由 UPnP 厂商规定。应大于 1800 秒（30 分钟）。关键字 “`second`”（秒）后为一个整数（无空格）或关键字 “`infinite`”（无限）。

若发布者不能接受新订阅，或是订阅请求存在错误，发布者必须返回以下一个错误进行响应。该响应必须在 30 秒（包括预计的传输时间）内发出。

错误

不兼容标头

400 Bad Request。如果 SID 标头和一个 NT 或 CALLBACK 标头一起出现，发布者必须返回 HTTP 错误 400 Bad Request 进行响应。

缺少或无效的 CALLBACK

412 Precondition Failed。如果 CALLBACK 标头缺少或不包含有效的 HTTP URL，发布者必须返回 HTTP 错误 412 Precondition Failed. 进行响应。

无效 NT

412 Precondition Failed。如果 NT 标题不符合 [upnp:event](#)，发布者必须返回 HTTP 错误 412 Precondition Failed.进行响应。

不能接收订阅

5xx。如果发布者不能接受订阅，它必须返回 HTTP 500 系列错误代码进行响应。

其它错误可能被 UPnP 下的协议栈层返回。参考这些协议的相关文档了解详细信息。

4.1.2 事件触发：续订：通过 SID 续订

为续订特定服务的事件，续订消息将被发往该服务的事件触发 URL。（注意：事件触发 URL 可能相对于基础 URL。）然而，与最初的订阅消息不同，续订消息既不包含服务标识符也不包含事件消息交付 URL，而是包含由发布者分配的订阅标识符，为续订提供明确的参考。与订阅消息相同，续订消息还可能包含所要求的订阅持续时间。

续订消息采用与订阅消息相同的方法，但这两种消息使用的是一组不相关的标头；续订消息使用 SID 标头，而订阅消息使用 NT 和 CALLBACK 标头。包含 SID 和 NT 或 CALLBACK 标头的消息是一个错误。

要续订服务事件，订阅者必须以下列格式发送一个采用 SUBSCRIBE 方法和 SID 标头的请求。用斜体表示的值为实际值的占位符。

```
SUBSCRIBE publisher path HTTP/1.1  
HOST: publisher host:publisher port  
SID: uuid:subscription UUID  
TIMEOUT: Second-requested subscription duration
```

（采用 SUBSCRIBE 方法的请求没有消息体，但请注意，该消息与最后一个 HTTP 标头之间必须空一行。）

以下列出了上述列表中出现的命令行和标头的详细信息。除特别注明外，所有标头值均要区分大小写。

命令行

SUBSCRIBE

方法由 GENA 定义。开始订阅或续订。

发布者路径

事件 URL（设备描述中 [服务](#) 元素的 [eventSubURL](#) 子元素）的路径组成。单一、相对 URL。

HTTP/1.1

HTTP 版本。

标头

HOST

要求。事件触发 URL（设备描述中 [服务](#) 元素的 [eventSubURL](#) 子元素）的域名或 IP 地址和可选端口组件。如果该缺少或为空，则假定为端口 80。

CALLBACK

（续订事件订阅不使用 CALLBACK 标头。）

NT

（续订事件订阅不使用 NT 标头。）

SID

GENA 规定使用的标头。订阅标识符。发布者必须指定订阅标识符来响应订阅请求。必须是唯一的。必须以 `uuid` 开头：由 UPnP 厂商规定。单一 URI。

TIMEOUT

推荐使用。直到订阅期满所要求的持续时间，秒数或无限时间。由 UPnP 论坛工作委员会推荐。由 UPnP 厂商规定。关键字 “second”（秒）后为一个整数（无空格）或关键字 “infinite”（无限）。

要接受续订，发布者需重新分配一个订阅持续时间，且必须以与响应新订阅请求相同的格式发送一个响应。（无初始化事件消息。）

若发布者不能接受续订，或是续订请求存在错误，发布者发送下列一个错误进行响应。该响应必须在 30 秒（包括预计的传输时间）内发出。

错误

不兼容标头

400 Bad Request。如果 SID 标头和一个 NT 或 CALLBACK 标头一起出现，发布者必须返回 HTTP 错误 400 Bad Request 进行响应。

无效 SID

412 Precondition Failed。如果 SID 没有响应一个未过期的已知订阅，则发布者必须返回 HTTP 错误 412 Precondition Failed.进行响应。

缺少 SID

412 Precondition Failed。如果 SID 标头缺少或为空，发布者必须返回 HTTP 错误 412 Precondition Failed.进行响应。

不能接受续订

5xx。如果发布者不能接受续订，则必须返回 HTTP 500 系列错误代码进行响应。

其它错误可能会被 UPnP 下的协议栈层返回。参考这些协议的相关文档了解详细信息。

4.1.3 事件：取消订阅：UNSUBSCRIBE

当不再需要来自特定服务的事件时，需要向该服务的事件触发 URL 发送取消消息。（注意：事件触发 URL 可能相对于基本 URL。）该信息包含订阅标识符。取消订阅通常可以减少服务、控制点和网络负载。如果一个控制点突然从网络上退出，它可能无法再发送取消消息。根据规定，除非续订否则订阅最终将会自行期满。

要取消服务事件的订阅，订阅者必须以下列格式发送一个采用 UNSUBSCRIBE 方法的请求。用斜体表示的值为实际值的占位符。

UNSUBSCRIBE *publisher path* HTTP/1.1

HOST: *publisher host.publisher port*

SID: *uuid:subscription UUID*

（采用 UNSUBSCRIBE 方法的请求没有消息体，但请注意，该消息与最后一个 HTTP 标头之间必须空一行。）

以下列出了上述列表中出现的命令行和标头的详细信息。除特别注明外，所有标头值均要区分大小写。

命令行

UNSUBSCRIBE

方法由 GENA 定义。取消订阅。

发布者路径

事件 URL（设备描述中 **服务** 元素的 **eventSubURL** 子元素）的路径组成。单一、相对 URL。

HTTP/1.1

HTTP 版本。

标头

HOST

要求。事件 URL（设备描述中 **服务** 元素的 **eventSubURL** 子元素）的域名或 IP 地址和可选端口组件。如果该端口缺少或为空，则假定为端口 80。

CALLBACK

（取消事件订阅不使用 CALLBACK 标头。）

NT

（取消事件订阅不使用 NT 标头。）

SID

GENA 规定使用的标头。订阅标识符。发布者必须分配订阅标识符来响应订阅请求。必须是唯一的。必须以 **uuid** 开头：由 UPnP 厂商规定。单一 URI。

TIMEOUT

（取消事件订阅不使用 TIMEOUT 标头。）

要取消订阅，发布者必须在 30 秒（包括预计的传输时间）内以下面的格式发送一个响应。

HTTP/1.1 200 OK

如果取消请求出现错误，设备必须返回以下一种错误进行响应。该响应必须在 30 秒（包括预计的传输时间）内发出。

错误

不兼容标头

400 Bad Request。如果 SID 标头和一个 NT 或 CALLBACK 标头一起出现，则发布者必须返回 HTTP 错误 400 Bad Request 进行响应。

无效 SID

412 Precondition Failed.如果 SID 没有响应一个未过期的已知订阅，则发布者必须返回 HTTP 错误 **412 Precondition Failed.**进行响应。

缺少 SID

412 Precondition Failed.如果 SID 标头缺少或为空，发布者必须返回 HTTP 错误 **412 Precondition Failed.**进行响应。

其它错误可能会被 UPnP 下的协议栈层返回。参考这些协议的相关文档了解详细信息。

4.2 事件：事件消息

服务通过发送事件消息来发布其状态变量的变更。这些消息包含一个或多个状态变量名称和这些变量的当前值。事件消息应尽可能快地发送，以使订阅者获得有关服务的准确信息，并支持订阅者显示响应用户界面。如果有一个以上变量的值同时发生改变，发布者应将这些变化捆绑在一个事件消息中，以减少处理和网络流量。

如上所述，在订阅者第一次订阅时，需要发送一个初始化事件消息。该事件消息包含所有事件触发变量的名称和值，并且允许订阅者初始化其服务状态模型。该消息应在发布者接受订阅后尽可能快地发送。

事件消息均标有事件编号。为便于进行错误检查，发布者必须保持每个订阅有一个单独的事件编号（解释如下）。当发布者发送初始化事件消息时，订阅事件编号被初始化为 0。对于以后的每条事件消息，发布者会增加订阅事件编号，包括事件消息更新的编号。任何事件编号的实施均需处理溢出，并将事件编号重置到 1（而不是 0）。订阅者还要在下一个事件编号不是前一个编号的增量的情况下，处理这一特例情况。应作为 4 字节（32 位）整数来实施。

如果订阅者未响应事件消息，发布者应继续向订阅者发送事件消息，直到订阅期满为止。

为修复事件订阅，例如，如果订阅者已丢失一条或多条事件消息，订阅者必须取消订阅并重新进行订阅。这样，订阅者将获得一个新的订阅标识符、一个新的初始化事件消息和一个新的事件编号。

4.2.1 事件：事件消息：NOTIFY

要发送事件消息，发布者必须以下列格式发送一个采用 NOTIFY 方法的请求。以下斜体表示的值为实际值的占位符。

```
NOTIFY delivery path HTTP/1.1
HOST: delivery host:delivery port
CONTENT-TYPE: text/xml
CONTENT-LENGTH: Bytes in body
NT: upnp:event
NTS: upnp:propchange
SID: uuid:subscription-UUID
SEQ: event key

<e:propertyset xmlns:e="urn:schemas-upnp-org:event-1-0">
  <e:property>
    <variableName>new value</variableName>
  </e:property>
  Other variable names and values (if any) go here.
</e:propertyset>
```

以下列出了上述列表中出现的命令行、标头和消息体元素的详细信息。除特别注明外，所有标头值均要区分大小写。所有消息体元素和属性均区分大小写；除特别注明外，消息体值不区分大小写。除特别注明外，元素的顺序无关紧要。除特别注明外，所要求的元素必须出现一次（无重复），推荐或可选元素最多只能出现一次。

命令行

NOTIFY

方法由 **GENA** 定义。通知客户关于事件的情况。

交付路径

交付 **URL**（订阅消息中的 **CALLBACK** 标头）的路径组成。事件消息的目的地。单一、相对 **URL**。

HTTP/1.1

HTTP 版本。

标头

HOST

要求。交付 **URL**（订阅消息中的 **CALLBACK** 标头）的域名或 **IP** 地址和可选端口组件。如果该缺少或为空，则假定为端口 **80**。

ACCEPT-LANGUAGE

（事件消息中未使用 **ACCEPT-LANGUAGE** 标头。）

CONTENT-LENGTH

要求。消息体长度以字节来表示，为整数。

CONTENT-TYPE

要求。必须为 **text/xml**。

NT

GENA 规定使用的标头。通知类型。必须采用 **upnp:event** 形式。

NTS

GENA 规定使用的标头。通知子类型。必须采用 **upnp:propchange** 形式。

SID

GENA 规定使用的标头。订阅标识符。必须是唯一的。必须以 **uuid** 开头：由 **UPnP** 厂商规定。单一 **URI**。

SEQ

UPnP 规定使用的标头。事件编号。初始化事件消息必须为 0。发送给特定订阅者的每条事件消息必须以 1 为增量。长度应为 8 字节。为防止溢出，必须回置到 1。一个整数。

消息体

propertyset

要求。Xmlns 名字空间属性必须为 urn : [schemas-upnp-org:event-1-0](#)。所有子元素必须符合这一名字空间。包含以下子元素：

特性

要求。事件消息中的变量名称和值重复一次。值必须符合 [propertyset](#) 名字空间。包含以下子元素：

variableName

要求。元素是更改的状态变量名称（服务描述中 [stateVariable](#) 元素的 [名字](#) 子元素）。值必须符合 [propertyset](#) 名字空间。值是指这一状态变量的新值。单一数据类型由 UPnP 服务描述确定。

为了获得未来可伸缩性，在处理上述 XML 时要遵守灵活 XML 处理框架（FXPP），设备和控制点必须忽略：(a) 任何未知元素及其子元素或内容，以及(b)任何未知属性及其值。

要确认收到了这一事件消息，订阅者必须要在 30 秒（包括预计的传输时间）内做出响应。如果订阅者未在 30 秒内做出响应，发布者应停止向订阅者发送此消息，但应保持订阅并向订阅者发送未来事件消息，直到订阅期满或取消为止。订阅者必须采用以下格式发送响应：

HTTP/1.1 200 OK

（采用 NOTIFY 方法的请求无消息体，但请注意，消息与最后一个 HTTP 标头之间必须空一行。）

如果事件消息存在错误，则订阅者必须返回下面一种错误进行响应。该响应必须在 30 秒（包括预计的传输时间）内发出。

错误

SID 缺少

412 Precondition Failed。如果 SID 标头缺少或为空，订阅者必须返回 HTTP 错误 412 Precondition Failed.进行响应。

无效 SID

412 Precondition Failed。如果 SID 不响应一个已知的订阅，订阅者必须返回 HTTP 错误 412 Precondition Failed.进行响应。（当服务接收到这一错误响应时，必须终止这一 SID。）

NT 或 NTS 标头缺少

400 Bad Request。如果 NT 或 NTS 标头缺少，订阅者必须返回 HTTP 错误 400 Bad Request 进行响应。

无效 NT 标头

412 Precondition Failed.如果 NT 标头不符合 [upnp:event](#)，订阅者必须返回 HTTP 错误 412 Precondition Failed.进行响应。

无效 NTS 标头

412 Precondition Failed.如果 NTS 标头不符合 [upnp:propchange](#)，订阅者必须返回 HTTP 错误 412 Precondition Failed.进行响应。

其它错误可能会被 UPnP 下的协议栈层返回。参考这些协议的相关文档了解详细信息。

4.3 事件：用于事件的 UPnP 模板语言

UPnP 模板语言定义了用于设备和服务的合适模板。从较小程度上讲，它还可提供事件消息消息体模板。描述部分介绍了与设备和服务相关的 UPnP 模板语言。如该部分所述，UPnP 模板语言以 XML 句法编写，源于 XML 模式（第 1 部分：结构，第 2 部分：数据类型）。以下是与事件相关的语言列表。它定义的元素用于事件消息；它们在此处用绿色表示，在以上列表中同样用绿色表示。下面是定义这些元素的地方（虽然是最低限度的定义）；上面是使用这些元素的地方。

紧随其后是对 XML 模式元素、属性和所用值的简单说明。本部分结尾关于 XML 模式的参考中还介绍了更详细的内容。

用于事件触发的 UPnP 模板语言

```
<?xml version="1.0" ?>
<Schema name="urn:schemas-upnp-org:event-1-0"
xmlns="urn:schemas-microsoft-com:xml-data"
xmlns:dt="urn:schemas-microsoft-com:datatypes">
  <ElementType name="propertyset" content="eltOnly">
    <element type="property" minOccurs="1" maxOccurs="*" />
  </ElementType>
  <ElementType name="property" content="eltOnly" />
</Schema>
```

元素

引用一个元素旨在表明嵌套。MaxOccurs 属性定义了元素必须出现的最多次数；缺省为 maxOccurs = 1；可显示一次或多次的元素具有 maxOccurs = *。

ElementType

用全新的衍生语言定义一个元素。名称属性定义元素名称。模式属性表明全新衍生语言中的元素是否包含此处没有明确规定的元素；当可能只包含未规定的子元素时，model = open。内容属性表明可能包含的内容；只包含其它元素的元素带有 content = eltOnly。

正如描述部分所述，服务的 UPnP 模板语言还规定了状态变量的 `sendEvents` 属性。这一属性的缺省值为 yes。要表示一个状态变量可以被事件触发，则在服务描述中该属性的值为 yes（或者该属性被省略）；要表示状态变量不能被事件触发，则属性值为 no。注意，如果服务的所有状态变量均不能被事件触发，则该服务就没有内容可以发布，且控制点不能订阅、也不会收到来自该服务的事件消息。

4.4 事件：增加 UPnP 模板语言

可以使用 UPnP 模板中没有涵盖的注释来增加设备和服务的描述。从较小程度上讲，这些注释中的值可以用来捕获事件过滤或事件适度。

如上所述，某些状态变量值的改变太快，以致事件触发功能没有意义。以下是面向 UPnP 论坛工作委员会或 UPnP 厂商的推荐词汇表，用于记录由于变量值改变而发出的事件消息数量适度。

maximumRate = *n*

可选。超过 *n* 秒频度的状态变量 *v* 不会成为事件消息的一部分。如果 *v* 是唯一变化的变量，那么事件消息的生成频度通常不会超过 *n* 秒。如果 *v* 在事件消息发出后的 *n* 秒内停止变化，则事件消息必须以新的 *v* 值发送。建议采用可模拟不断变化的属性的变量。一个整数。

minimumDelta = *n*

可选。状态变量 *v* 不会成为事件消息的一部分，除非它的值自上次包含 *v* 的事件消息发送后改变超过 $n * \text{allowedValueRange}$ 步，例如，除非 *v* 已增加了 *n* 倍。（控制部分介绍的内容参见：INCREMENT、INCREMENT_BOUNDED 和 INCREMENT_WRAP。）仅定义了带编号的变量和实际数据类型。建议采用模拟计数器的变量。一个整数。

发布者可以在事件发出时发送任何更改的调节变量。发布者应努力尝试满足上述适度规则，但可以在发出事件后刷新最近的变化。

注意适度仅对事件有影响，而不会影响到状态表的更新。具体而言，**QueryStateVariable** 可以通过事件触发返回一个比已发布的值更新的值。换言之，适度意味着不是所有状态表变化都会导致事件发生。

关于哪些变量用于事件触发及任何可能的调整，由适当的 UPnP 论坛工作委员会（面向标准服务）或 UPnP 厂商（面向非标准服务）来决定。

4.5 事件参考

FXPP

灵活的 XML 处理框架。规定必须忽略未知的 XML 元素及其子元素。
IETF 草案。

GENA

通用事件通知架构。IETF 草案。

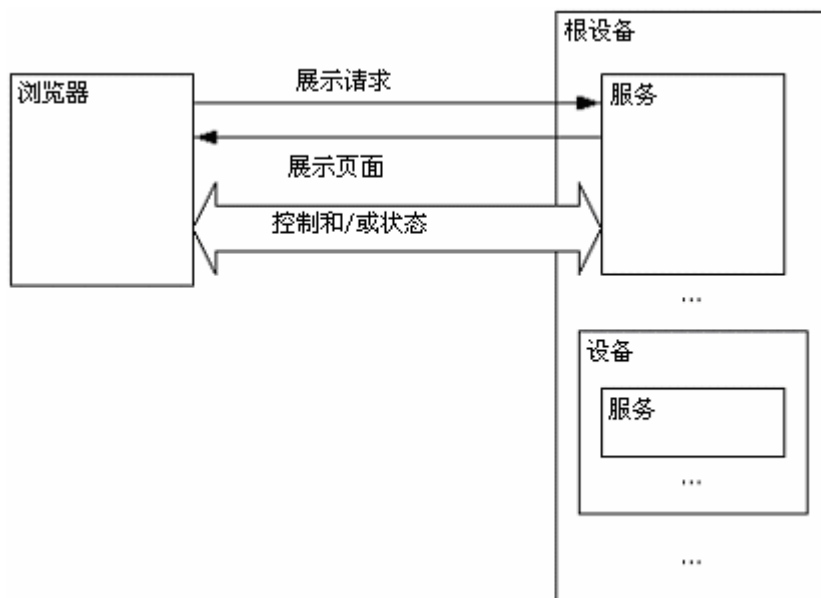
XML 模式（第 1 部分：结构，第 2 部分：数据类型）。

语法定义 UPnP 模板语言。使用 XML 定义。W3C 工作草案。第 1 部分：结构。第 2 部分：数据类型。

5. 展示

展示是 UPnP 网络的第 5 步。首先“寻址”（第 0 步）设备获得一个网络地址，之后“发现”（第 1 步）控制点发现新设备，然后“描述”（第 2 步）控制点了解该设备能力。展示展示了一个基于 HTML 的用户界面，用以控制和/或浏览设备状态。展示是“控制”（第 3 步：控制点向设备发出动作）和“事件触发”（第 4 步：控制点监听设备的状态变化）的补充。

在控制点(1)发现设备和(2)取得设备描述之后，控制点即准备开始提供展示。如果设备拥有进行展示的 URL，那么控制点就可以通过此 URL 取得一个页面，在浏览器中加载该页面，并根据页面功能，支持用户控制设备和/或浏览设备状态。每一项完成的程度取决于展示页面和设备的具体功能。



展示 URL 包含在设备描述的 presentationURL 元素中。设备描述通过描述消息提供。描述部分就设备描述和描述信息做了详细阐述。

取得展示页面是一个基于 HTTP 的简单过程，使用整体 UPnP 协议栈的以下子集。（总体 UPnP 协议栈在本文开始列出。）

UPnP 厂商[紫色]
UPnP 设备架构[绿色]
HTTP[黑色]
TCP[黑色]
IP[黑色]

在最高层，展示页面由 UPnP 厂商规定。沿着协议栈向下，UPnP 设备架构规定该页面以 HTML 编写。该页面通过 TCP/IP 层上的 HTTP 层提供。为便于参考，包括[方括号]中的颜色，以与本文其它部分保持一致。

为获得展示页面，控制点发出一个 HTTP GET 请求到展示 URL，设备返回一个展示页面。

与 UPnP 设备和服务模板、以及标准设备和服务类型不同，展示页面的功能完全由 UPnP 厂商规定。展示页面并非由 UPnP 论坛工作委员会提供支持。页面必须是 HTML 页面；应为 HTML 3.0 或更高版。其它设计方面均由厂商规定。这包括但不限于控制点浏览器的所有功能、所使用的脚本语言或浏览器插件、以及与设备进行交互的方式。为实现展示页面，UPnP 厂商可能希望使用 UPnP 机制来控制/或触发事件，以充分利用设备现有的功能，但并不强制规定必须这样做。

展示页面应使用 HTML 中提供的机制进行本地化（如带有 charset 属性的 META 标记）。控制点应使用 HTTP 的 ACCEPT- / CONTENT-LANGUAGE 特性，尽可能地取得本地化的展示页面。具体来讲，控制点可能在展示页面请求中包含一个 HTTP ACCEPT-LANGUAGE 标头；如果请求中出现了一个 ACCEPT-LANGUAGE 标头，响应必须要包含一个 CONTENT-LANGUAGE 标头，以确定该页面的语言。

5.1 展示参考

HTML

超文本标记语言。W3C 推荐。<<<http://www.w3.org/MarkUp/>>>

动作

服务发出的命令。包含一个或多个输入或输出变量。可能具有一个返回值。如欲了解更多信息，请参阅描述和控制部分。

变量

服务所发出的动作参数。可以采用 `in xor out` 格式。如欲了解更多信息，请参阅描述和控制部分。

控制点

取得设备和服务描述、向服务发送动作、进行服务状态变量轮询和接收来自服务的事件。

设备

逻辑设备。一个容器。可以嵌入其它逻辑设备。嵌入一个或多个服务。如欲了解更多信息，请参阅描述部分。

设备描述

逻辑设备的正式定义，采用 UPnP 模板语言表达。以 XML 句法编写。由 UPnP 厂商通过填写 UPnP 设备模板中的占位符来指定，包括：制造商名称、型号名称、型号、序列号、以及控制、事件触发和展示的 URL。如欲了解更多信息，请参阅描述部分。

设备类型

标准设备类型由 `urn:schemas-upnp-org:device:`来指定，后面带有 UPnP 论坛工作委员会指定的唯一名称来表示。与 UPnP 设备模板为一对一的关系。UPnP 厂商可以指定其它设备类型；这些由 `urn:domain-name:device:`指定。后面带有厂商指定的唯一名称来表示，其中域名为厂商注册域名。如欲了解更多信息，请参阅描述部分。

事件

服务发出的状态变量的一个或多个变化的通知。如欲了解更多信息，请参阅事件部分。

发布者

事件消息来源。通常为设备服务。如欲了解更多信息，请参阅事件部分。

根设备

不嵌入任何其它逻辑设备的设备。如欲了解更多信息，请参阅描述部分。

服务

逻辑功能单位。控制的最小单位。暴露状态并使用状态变量来模拟物理设备的状态。如欲了解更多信息，请参阅控制部分。

服务描述

逻辑服务的正式定义，采用 UPnP 模板语言表达。以 XML 句法编写。由 UPnP 厂商填写 UPnP 服务模板（原为 SCPD）中的所有占位符来指定。如欲了解更多信息，请参阅描述部分。

服务类型

标准服务类型由 `urn:schemas-upnp-org:service:`指定。后面带有 UPnP 论坛工作委员会指定的唯一名称和一个整数版本号码来表示。与 UPnP 服务模板为一对一的关系。UPnP 厂商可以指定其它的服务；由 `urn:domain-name:service:`指定。后面带有的厂商指定的唯一名称、冒号和一个版本号来表示，其中 *域名*为厂商注册域名。如欲了解更多信息，请参阅描述部分。

SOAP

简单对象访问协议。一种基于 XML 的远程调用机制，通过 HTTP 发送命令和接收值。如欲了解更多信息，请参阅控制部分。

SSDP

简单服务发现协议。一种使用基于 UDP 的 HTTP 协议多播变量的多播发现和搜索机制。如欲了解更多信息，请参阅发现部分。

状态变量

一个物理服务的模型的一个方面。由服务发出。具有名称、数据类型、可选缺省值、可选限制值、并可在该值改变时触发事件。如欲了解更多信息，请参阅描述和控制部分。

订阅者

事件消息接收方。通常为控制点。如欲了解更多信息，请参阅事件部分。

UPnP 设备模板

模板列出设备类型、所要求的嵌入式设备（如有）和所要求的服务。以 XML 句法编写，源自 UPnP 模板语言。由 UPnP 论坛工作委员会定义。与标准设备类型为一对一的关系。如欲了解更多信息，请参阅描述部分。

UPnP 服务模板

模板列出动作名称、这些动作的参数、状态变量和这些状态变量的属性。以 XML 句法编写，源自 UPnP 模板语言。由 UPnP 论坛工作委员会定义。与标准服务类型为一对一的关系。如欲了解更多信息，请参阅描述部分。

UPnP 模板语言

定义了 UPnP 设备和服务模板中使用的元素和属性。以 XML 句法编写，源自 XML 模式（第 1 部分：结构，第 2 部分：数据类型）。由此处的 UPnP 设备架构定义。如欲了解更多信息，请参阅描述部分。

文件结束

文章翻译免责声明

本UPnP文档最初用英文发布，并仅有英文版本通过了UPnP论坛的正式审核。文章已采用翻译服务和翻译技术在英文版本的基础上翻译成目标语言版本，UPnP Implementers Corporation及其相关机构不对翻译版本做出任何保证，也不为由翻译不准确所导致的直接或间接损失承担责任。在使用翻译版本中所包括的技术信息时，用户同意UPnP Implementers Corporation和UPnP论坛成员对于英文到目标语言翻译的不完整、或不准确导致的全部或部分损失不承担任何责任。此外，用户同意应保护UPnP Implementers Corporation免受由语言翻译而带来的伤害。