Universal Plug and Play Device Architecture

Version 1.0, 08 Jun 2000 10:41 AM .

Table of contents

Introduction

What is Universal Plug and Play?

Universal Plug and Play is an architecture for pervasive peer-to-peer network connectivity of intelligent appliances, wireless devices, and PCs of all form factors. It is designed to bring easy-to-use, flexible, standards-based connectivity to ad-hoc or unmanaged networks whether in the home, in a small business, public spaces, or attached to the Internet. Universal Plug and Play is a distributed, open networking architecture that leverages TCP/IP and the Web technologies to enable seamless proximity networking in addition to control and data transfer among networked devices in the home, office, and public spaces.

UPnP is more than just a simple extension of the plug and play peripheral model. It is designed to support zero-configuration, "invisible" networking, and automatic discovery for a breadth of device categories from a wide range of vendors. This means a device can dynamically join a network, obtain an IP address, convey its capabilities, and learn about the presence and capabilities of other devices. DHCP and DNS servers are optional and are used only if available on the network. Finally, a device can leave a network smoothly and automatically without leaving any unwanted state behind.

UPnP leverages Internet components, including IP, TCP, UDP, HTTP, and XML. Like the Internet, contracts are based on wire protocols that are declarative, expressed in XML, and communicated via HTTP. IP internetworking is a strong choice for UPnP because of its proven ability to span different physical media, to enable real world multiple-vendor interoperation, and to achieve synergy with the Internet and many home and office intranets. UPnP has been explicitly designed to accommodate these environments. Further, via bridging, UPnP accommodates media running non-IP protocols when cost, technology, or legacy prevents the media or devices attached to it from running IP.

What is "universal" about UPnP? No device drivers; common protocols are used instead. UPnP networking is media independent. UPnP devices can be implemented using any programming language, and on any operating system. UPnP does not specify or constrain the design of an API for applications running on control points; OS vendors may create APIs that suit their customer's needs. UPnP enables vendor control over device UI and interaction using the browser as well as conventional application programmatic control.

UPnP Forum

The UPnP Forum is an industry initiative designed to enable easy and robust connectivity among stand-alone devices and PCs from many different vendors. The UPnP Forum seeks to develop standards for describing device protocols and XML-based device schemas for the purpose of enabling device-to-device interoperability in a scalable networked environment. The UPnP Forum oversees a logo program for compliant devices.

The UPnP Forum has set up working committees in specific areas of domain expertise. These working committees are charged with

creating proposed device standards, building sample implementations, and building appropriate test suites. This document indicates specific technical decisions that are the purview of UPnP Forum working committees.

UPnP vendors can build compliant devices with confidence of interoperability and benefits of shared intellectual property and the logo program. Separate from the logo program, vendors may also build devices that adhere to the UPnP Device Architecture defined herein without a formal standards procedure. If vendors build non-standard devices, they determine technical decisions that would otherwise be determined by a UPnP Forum working committee.

In this document

The Universal Plug and Play Device Architecture (formerly known as the DCP Framework) contained herein defines the protocols for communication between controllers, or *control points*, and devices. For discovery, description, control, eventing, and presentation, UPnP uses the following protocol stack.

<table>
<tr><td colspan="6">UPnP vendor [purple]</td></tr>
<tr><td colspan="6">UPnP Forum [red]</td></tr>
<tr><td colspan="6">UPnP Device Architecture [green]</td></tr>
<tr><td colspan="2">HTTPMU (multicast) [black]</td><td>GENA [navy]</td><td>SSDP [blue]</td><td>HTTPU (unicast) [black]</td><td>SSDP [blue]</td></tr>
<tr><td colspan="4" rowspan="2"></td><td>SOAP [blue]
HTTP [black]</td><td>HTTP [black]</td><td>GENA [navy]</td></tr>
<tr><td colspan="4">UDP [black]</td><td colspan="2">TCP [black]</td></tr>
<tr><td colspan="6">IP [black]</td></tr>
</table>

At the highest layer, messages logically contain only UPnP vendor-specific information about their devices. Moving down the stack, vendor content is supplemented by information defined by UPnP Forum working committees. Messages from the layers above are hosted in UPnP-specific protocols, defined in this document. In turn, the above messages are formatted using the Simple Service Discovery Protocol (SSDP), General Event Notification Architecture (GENA), and Simple Object Access Protocol (SOAP). The above messages are delivered via HTTP, either a multicast or unicast variety running over UDP, or the standard HTTP running over TCP. Ultimately, all messages above are delivered over IP. The remaining sections of this document describe the content and format for each of these protocol layers in detail. For reference, colors in [square brackets] above indicate which protocol defines specific message components throughout this document.

The foundation for UPnP networking is IP addressing. Each device must have a Dynamic Host Configuration Protocol (DHCP) client and search for a DHCP server when the device is first connected to the network. If a DHCP server is available, i.e., the network is managed, the device must use the IP addressed assigned to it. If no DHCP server is available, i.e., the network is unmanaged, the device must use Auto IP to get an address. In brief, Auto IP defines how a device intelligently chooses an IP address from a set of reserved addresses and is able to move easily between managed and unmanaged networks. If during the DHCP transaction, the device obtains a domain name, e.g., through a DNS server or via DNS forwarding, the device should use that name in subsequent network operations; otherwise, the device should use its IP address.

Given an IP address, Step 1 in UPnP networking is discovery. When a device is added to the network, the UPnP discovery protocol allows that device to advertise its services to control points on the network. Similarly, when a control point is added to the network, the UPnP discovery protocol allows that control point to search for devices of interest on the network. The fundamental exchange in both cases is a discovery message containing a few, essential specifics about the device or one of its services, e.g., its type, identifier, and a pointer to more detailed information. The UPnP discovery protocol is based on the Simple Service Discovery Protocol (SSDP). The section on Discovery below explains how devices advertise, how control points search, and details of the format of discovery messages.

Step 2 in UPnP networking is description. After a control point has discovered a device, the control point still knows very little about the device. For the control point to learn more about the device and its capabilities, or to interact with the device, the control point must retrieve the device's description from the URL provided by the device in the discovery message. Devices may contain other, logical devices, as well as functional units, or *services*. The UPnP description for a device is expressed in XML and includes vendor-specific, manufacturer information like the model name and number, serial number, manufacturer name, URLs to vendor-specific Web sites, etc. The description also includes a list of any embedded devices or services, as well as URLs for control, eventing, and presentation. For each service, the description includes a list of the commands, or *actions*, the service responds to, and parameters, or *arguments*, for each action; the description for a service also includes a list of variables; these variables model the state of the service at run time, and are described in terms of their data type, range, and event characteristics. The section on Description below explains how devices are described and how those descriptions are retrieved by control points.

Step 3 in UPnP networking is control. After a control point has retrieved a description of the device, the control point can send actions to a device's service. To do this, a control point sends a suitable control message to the control URL for the service (provided in the device description). Control messages are also expressed in XML using the Simple Object Access Protocol (SOAP). Like function calls, in response to the control message, the service returns any action-specific values. The effects of the action, if any, are modeled by changes in the variables that describe the run-time state of the service. The section on Control below explains the description of actions, state variables, and the format of control messages.

Step 4 in UPnP networking is eventing. A UPnP description for a service includes a list of actions the service responds to and a list of variables that model the state of the service at run time. The service publishes updates when these variables change, and a control point may subscribe to receive this information. The service publishes updates by sending event messages. Event messages contain the names of one of more state variables and the current value of those variables. These messages are also expressed in XML and formatted using the General Event Notification Architecture (GENA). A special initial event message is sent when a control point first subscribes; this event message contains the names and values for all evented variables and allows the subscriber to initialize its model of the state of the service. To support scenarios with multiple control points, eventing is designed to keep all control points equally informed about the effects of any action. Therefore, all subscribers are sent all event messages, subscribers receive event messages for all evented variables that have changed, and event messages are sent no matter why the state variable changed (either in response to a requested action or because the state the service is modeling changed). The section on Eventing below explains subscription and the format of event messages.

Step 5 in UPnP networking is presentation. If a device has a URL for presentation, then the control point can retrieve a page from this URL, load the page into a browser, and depending on the capabilities of the page, allow a user to control the device and/or view device status. The degree to which each of these can be accomplished depends on the specific capabilities of the presentation page and device. The section on Presentation below explains the protocol for retrieving a presentation page.

Audience

The audience for this document includes UPnP device vendors, members of UPnP Forum working committees, and anyone else who has a need to understanding the technical details of UPnP protocols.

This document assumes the reader is familiar with the HTTP, TCP, UDP, IP family of protocols; this document makes no attempt to explain them. This document also assumes most readers will be new to XML, and while it is not an XML tutorial, XML-related issues are addressed in detail given the centrality of XML to UPnP. This document makes no assumptions about the reader's understanding of various programming or scripting languages.

Required vs. recommended

In this document, features are described as Required, Recommended, or Optional as follows:

Required (or Must).
: These basic features must be implemented to comply with UPnP.

Recommended (or Should).
: These features add functionality supported by UPnP and should be implemented. Recommended features take advantage of the capabilities UPnP, usually without imposing major cost increases. Notice that for compliance testing, if a recommended feature is implemented, it must meet the specified requirements to be in compliance with these guidelines. Some recommended features could become requirements in the future.

Optional (or May).
: These features are neither required nor recommended by UPnP, but if the feature is implemented, it must meet the specified requirements to be in compliance with these guidelines. These features are not likely to become requirements in the future.

Acronyms

Acronym	Meaning	Acronym	Meaning
ARP	Address Resolution Protocol	SSDP	Simple Service Discovery Protocol
DHCP	Dynamic Host Configuration Protocol	UPC	Universal Product Code
DNS	Domain Name System	UPnP	Universal Plug and Play
FXPP	Flexible XML Processing Profile	URI	Uniform Resource Identifier
GENA	General Event Notification Architecture	URL	Uniform Resource Locator
HTML	HyperText Markup Language	URN	Uniform Resource Name
HTTPMU	HTTP Multicast over UDP	UUID	Universally Unique Identifier

HTTPU	HTTP (unicast) over UDP	XML	Extensible Markup Language
ICANN	Internet Corporation for Assigned Names and Numbers		
SOAP	Simple Object Access Protocol		

References and resources

RFC 2616
HTTP: Hypertext Transfer Protocol 1.1. IETF request for comments. <http://search.ietf.org/rfc/rfc2616.txt?number=2616>.
RFC 2279
UTF-8, a transformation format of ISO 10646 (character encoding). IETF request for comments. <http://search.ietf.org/rfc/rfc2279.txt?number=2279>.
XML
Extensible Markup Language. W3C recommendation. <http://www.w3.org/XML/>.

Each section in this document contains additional information about resources for specific topics.

Acknowledgments

The UPnP team at Microsoft gratefully acknowledges the help we received from the many technical reviewers representing the UPnP Forum Steering Committee and the UPnP vendor community. These reviewers contributed technical information, insights, and wisdom in developing this document.

We are also grateful to the software engineers, testers, and program managers at Microsoft who contributed feedback and technical content to ensure that the information in this document is accurate and timely.

0. Addressing

Addressing is Step 0 of UPnP networking. Through addressing, devices get a network address. Addressing enables discovery (Step 1) where control points find interesting device(s), description (Step 2) where where control points learn about device capabilities, control (Step 3) where a control point sends commands to device(s), eventing (Step 4) where control points listen to state changes in device(s), and presentation (Step 5) where control points display a user interface for device(s).

The foundation for UPnP networking is IP addressing. Each device must have a Dynamic Host Configuration Protocol (DHCP) client and search for a DHCP server when the device is first connected to the network. If a DHCP server is available, i.e., the network is managed, the device must use the IP addressed assigned to it. If no DHCP server is available, i.e., the network is unmanaged; the device must use automatic IP addressing (Auto-IP) to obtain an address.

Auto-IP defines how a device: (a) determines if DHCP is unavailable, and (b) intelligently chooses an IP address from a set of link-local IP addresses. This method of address assignment enables a device to easily move between managed and unmanaged networks.

The operations described in this section are further clarified in the reference documents listed below. Where conflicts between this document and the reference documents exist, the reference document always takes precedence.

0.1 Addressing: Determining whether to use Auto-IP

A device that supports AUTO-IP and is configured for dynamic address assignment begins by requesting an IP address via DHCP by sending out a DHCPDISCOVER message. The amount of time this DHCP Client should listen for DHCPOFFERS is implementation dependent. If a DHCPOFFER is received during this time, the device must continue the process of dynamic address assignment. If no valid DHCPOFFERS are received, the device may then auto-configure an IP address.

0.2 Addressing: Choosing an address

To auto-configure an IP address using Auto-IP, the device uses an implementation dependent algorithm for choosing an address in the 169.254/16 range. The first and last 256 addresses in this range are reserved and must not be used.

The selected address must then be tested to determine if the address is already in use. If the address is in use by another device, another address must be chosen and tested, up to an implementation dependent number of retries. The address selection should be randomized to avoid collision when multiple devices are attempting to allocate addresses.

0.3 Addressing: Testing the address

To test the chosen address, the device must use an Address Resolution Protocol (ARP) probe. An ARP probe is an ARP request with the device hardware address used as the sender's hardware address and the sender's IP address set to 0s. The device will then listen for responses to the ARP probe, or other ARP probes for the same IP address. If either of these ARP packets is seen, the device must consider the address in use and try a new address.

0.4 Addressing: Periodic checking for dynamic address availability

A device that has auto-configured an IP address must periodically check for the existence of a DHCP server. This is accomplished by sending DHCPDISCOVER messages. How often this check is made is implementation dependent, but checking every 5 minutes would maintain a balance between network bandwidth required and connectivity maintenance. If a DHCP offer is received, the device must proceed with dynamic address allocation. Once a DHCP assigned address is in place, the device may release the auto-configured address, but may also choose to maintain this address for a period of time to maintain connectivity.

To switch over from one IP address to a new one, the device must cancel any outstanding advertisements and reissue new ones. The section on Discovery explains advertisements and their cancellations.

0.5 Addressing: Device naming and DNS interaction

Once a device has a valid IP address for the network, it can be located and referenced on that network through that address. There may be situations where the end user needs to locate and identify a device. In these situations, a friendly name for the device is much easier for a human to use than an IP address.

Moreover, names are much more static than IP addresses. Clients referring a device by name don't require any modification when IP address of a device changes. Mapping of the device's DNS name to its IP address could be entered into DNS database manually or dynamically according to RFC 2136. While computers and devices supporting dynamic DNS updates can register their DNS records directly in DNS, it is also possible to configure a DHCP server to register DNS records on behalf of these DHCP clients.

0.6 Addressing: Name to IP address resolution

A computer that needs to contact a device identified by a DNS name needs to discover its IP address. The computer submits a DNS query according to RFC1034 and 1035 to the pre-configured DNS server(s) and receives a response from a DNS server containing the IP address of the target device. A computer can be statically pre-configured with the list of DNS servers. Alternatively a computer could be configured with the list of DNS server through DHCP, or after the address assignment through a DHCPINFORM message.

0.7 Addressing references

Auto-IP
: Automatically Choosing an IP Address in an Ad-Hoc IPv4 Network. IETF draft. <http://search.ietf.org/internet-drafts/draft-ietf-dhc-ipv4-autoconfig-05.txt>.

RFC1034
: Domain Names - Concepts and Facilities. IETF request for comments. <http://search.ietf.org/rfc/rfc1034.txt?number=1034>.

RFC1035
: Domain Names - Implementation and Specification. IETF request for comments. <http://search.ietf.org/rfc/rfc1035.txt?number=1035>.

RFC 2131
: Dynamic Host Configuration Protocol. IETF request for comments. <http://search.ietf.org/rfc/rfc2131.txt?number=2131>.

RFC 2136
: Dynamic Updates in the Domain Name System. IETF request for comments. <http://search.ietf.org/rfc/rfc2136.txt?number=2136>.

Dynamic DNS Updates by DHCP Clients and Servers
: Interaction between DHCP and DNS. IETF Draft. <http://search.ietf.org/internet-drafts/draft-ietf-dhc-dhcp-dns-12.txt>.

1. Discovery

Discovery is Step 1 in UPnP networking. Discovery comes after addressing (Step 0) where devices get a network address. Through discovery, control points find interesting device(s). Discovery enables description (Step 2) where control points learn about device capabilities, control (Step 3) where a control point sends commands to device(s), eventing (Step 4) where control points listen to state

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.