[MS-PNM]:
People Near Me (PNM) Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft Open Specification Promise or the Community Promise. If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **Trademarks.**The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names.** The example companies, organizations, products, domain names, email addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications do not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them. Certain Open Specifications are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

Revision Summary

Date	Revision History	Revision Class	Comments
04/08/2008	0.1		Initial availability
05/16/2008	0.1.1	Editorial	Revised and edited the technical content.
06/20/2008	0.1.2	Editorial	Revised and edited the technical content.
07/25/2008	0.1.3	Editorial	Revised and edited the technical content.
08/29/2008	0.1.4	Editorial	Revised and edited the technical content.
10/24/2008	0.1.5	Editorial	Revised and edited the technical content.
12/05/2008	0.2	Minor	Updated the technical content.
01/16/2009	0.2.1	Editorial	Revised and edited the technical content.
02/27/2009	0.2.2	Editorial	Revised and edited the technical content.
04/10/2009	0.2.3	Editorial	Revised and edited the technical content.
05/22/2009	0.2.4	Editorial	Revised and edited the technical content.
07/02/2009	0.2.5	Editorial	Revised and edited the technical content.
08/14/2009	0.2.6	Editorial	Revised and edited the technical content.
09/25/2009	0.3	Minor	Updated the technical content.
11/06/2009	0.3.1	Editorial	Revised and edited the technical content.
12/18/2009	1.0	Major	Updated and revised the technical content.
01/29/2010	1.1	Minor	Updated the technical content.
03/12/2010	1.2	Minor	Updated the technical content.
04/23/2010	1.2.1	Editorial	Revised and edited the technical content.
06/04/2010	1.2.2	Editorial	Revised and edited the technical content.
07/16/2010	1.3	Minor	Clarified the meaning of the technical content.
08/27/2010	1.3	No change	No changes to the meaning, language, or formatting of the technical content.
10/08/2010	1.3	No change	No changes to the meaning, language, or formatting of the technical content.
11/19/2010	1.3	No change	No changes to the meaning, language, or formatting of the technical content.
01/07/2011	1.3	No change	No changes to the meaning, language, or formatting of

Date	Revision History	Revision Class	Comments
			the technical content.
02/11/2011	1.3	No change	No changes to the meaning, language, or formatting of the technical content.
03/25/2011	1.3	No change	No changes to the meaning, language, or formatting of the technical content.
05/06/2011	2.0	Major	Significantly changed the technical content.
06/17/2011	2.1	Minor	Clarified the meaning of the technical content.
09/23/2011	2.1	No change	No changes to the meaning, language, or formatting of the technical content.
12/16/2011	2.1	No change	No changes to the meaning, language, or formatting of the technical content.
03/30/2012	2.1	No change	No changes to the meaning, language, or formatting of the technical content.
07/12/2012	2.1	No change	No changes to the meaning, language, or formatting of the technical content.
10/25/2012	2.1	No change	No changes to the meaning, language, or formatting of the technical content.
01/31/2013	2.1	No change	No changes to the meaning, language, or formatting of the technical content.

Contents

1 Introduction

This is a specification of the People Near Me (PNM) Protocol.

PNM specifies a protocol for broadcasting and retrieving information concerning the presence or absence of people on a subnet, including whether they are online, busy, or away. It provides a service to identify people who are using computers and to allow those people to send and receive invitations to participate in programs that are installed on their computers.

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in RFC 2119. Sections 1.5 and 1.9 are also normative but cannot contain those terms. All other sections and examples in this specification are informative.

1.1 Glossary

The following terms are defined in [MS-GLOS]:

big-endian
endpoint
globally unique identifier (GUID)
Internet Protocol version 6 (IPv6)
little-endian
SOAP
Transmission Control Protocol (TCP)
Unicode
universally unique identifier (UUID)
User Datagram Protocol (UDP)
UTF-8
Web Services Description Language (WSDL)
XML
XML namespace
XML schema (XSD)

The following terms are specific to this document:

base64: An encoding scheme for the representation of binary data, which uses an alphabet of 64 characters [RFC4648] and is used as the default mechanism for authenticating clear text between a web browser and a web server.

endpoint: A tuple composed of an IP address, a port, and a protocol number, which uniquely identifies a communication endpoint.

endpoint friendly name: A **friendly name** that identifies a network **endpoint** of an application discovered using PNM.

endpoint reference: A mechanism that dynamically identifies and describes the addresses of service **endpoints** and instances for a resource in the **SOAP** header of a message. An **endpoint** reference may contain a number of individual properties that are required to identify the entity or resource being conveyed.

expired: A peer that has stopped participating in the PNM protocol without sending a **Bye** message.

friendly name: The human-readable name that is used to identify a computer on a network.

host byte order: The byte order used to represent multi-byte integer values that are stored on a host system. This order is **little-endian** on systems that use Intel processors.

multicast: A delivery method in which content is transmitted from a media server to multiple clients. The clients have no connection with the server. Instead, the server sends a single copy of the stream across the network to multicast-enabled routers, which then replicate the data. Clients can then receive the stream by monitoring a specific multicast IP address and port.

network byte order: The byte order used to represent multi-byte integer values that are sent from one system to another over a network. This order is **big-endian** on **TCP**/IP networks.

peer-to-peer (P2P): A collaboration technology characterized by dynamically formed, self-organized and self-managed, robust, serverless communication peer networks, which are sometimes referred to as "meshes". Each P2P node is identified by a unique (within the mesh) peer ID, and shares bidirectional connections with at least its nearest neighbors.

People Near Me friendly name: A **friendly name** that identifies the user of the system or higher level application on a machine discovered using PNM.

transport address (XAddr): An address that is used for communication between a client and a service. It is an element of **Probe Match** and **Resolve Match** messages in the Web Services Dynamic Discovery Protocol [WS-Discovery].

unicast: A delivery method used by media servers for providing content to connected clients, in which each client receives a discrete stream that no other client has access to.

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as described in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

References to Microsoft Open Specifications documentation do not include a publishing year because links are to the latest version of the documents, which are updated frequently. References to other documents include a publishing year when one is available.

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

[MS-DTYP] Microsoft Corporation, "Windows Data Types".

[RFC4122] Leach, P., Mealling, M., and Salz, R., "A Universally Unique Identifier (UUID) URN Namespace", RFC 4122, July 2005, http://www.ietf.org/rfc/rfc4122.txt

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, October 2006, http://www.ietf.org/rfc/rfc4648.txt

[SOAP1.2-1/2007] Gudgin, M., Hadley, M., Mendelsohn, N., et al., "SOAP Version 1.2 Part 1: Messaging Framework (Second Edition) ", W3C Recommendation 27, April 2007, http://www.w3.org/TR/2007/REC-soap12-part1-20070427/

[SOAP-UDP] Combs, H., Justice, J., Kakivaya, G., et al., "SOAP-over-UDP", September 2004, http://specs.xmlsoap.org/ws/2004/09/soap-over-udp/soap-over-udp.pdf

If you have any trouble finding [SOAP-UDP], please check here.

[WS-Addr-Core] World Wide Web Consortium, "Web Services Addressing 1.0 - Core", W3C Recommendation, May 2006, http://www.w3.org/TR/ws-addr-core/

[WS-Discovery] Beatty, J., Kakivaya, G., Kemp D., et al., "Web Services Dynamic Discovery (WS-Discovery)", April 2005, http://specs.xmlsoap.org/ws/2005/04/discovery/ws-discovery.pdf

If you have any trouble finding [WS-Discovery], please check here.

[WSDL] Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S., "Web Services Description Language (WSDL) 1.1", W3C Note, March 2001, http://www.w3.org/TR/2001/NOTE-wsdl-20010315

[XMLNS] Bray, T., Hollander, D., Layman, A., et al., Eds., "Namespaces in XML 1.0 (Third Edition)", W3C Recommendation, December 2009, http://www.w3.org/TR/2009/REC-xml-names-20091208/

[XMLSCHEMA1] Thompson, H.S., Beech, D., Maloney, M., and Mendelsohn, N., Eds., "XML Schema Part 1: Structures", W3C Recommendation, May 2001, http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/

[XMLSCHEMA2] Biron, P.V., and Malhotra, A., Eds., "XML Schema Part 2: Datatypes", W3C Recommendation, May 2001, http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.rfc-editor.org/rfc/rfc2119.txt

1.2.2 Informative References

This document references the following archived documents. For a list of archived documents, see Windows Archived Protocols.

- [MS-P2PPI] Microsoft Corporation, "Peer-to-Peer Presence and Invitation Protocol".

This document references the following documents:

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary".

[MSDN-P2PC] Microsoft Corporation, "Peer to Peer Collaboration Infrastructure", http://msdn.microsoft.com/en-us/library/aa371044.aspx

1.3 Overview

The People Near Me (PNM) Protocol is a **peer-to-peer (P2P)** protocol that is used to locate networked hosts or devices implementing PNM. PNM provides a way for peers to announce their presence and to locate each other in directly connected subnets.

PNM is a specialization of the Web Services Dynamic Discovery Protocol [WS-Discovery] and follows its model for announcing and locating resources. In PNM, each peer fulfills both the client and server roles that are defined in [WS-Discovery].

A peer announces its presence to connected IP subnets through the **multicast** of a **Hello** message on the local network. Peers discover other peers passively by listening for such messages. Peers can also solicit other peers by multicasting a **Probe** message that contains characteristics; peers with matching characteristics reply with **unicast** messages.

1.4 Relationship to Other Protocols

The WS-Discovery Protocol [WS-Discovery] and, therefore, PNM uses SOAP-over-UDP [SOAP-UDP] as a network transport.

1.5 Prerequisites/Preconditions

None.

1.6 Applicability Statement

The primary purpose of PNM is to locate peers hosted on the same subnet. PNM is not designed to locate peers that are hosted on other subnets.

1.7 Versioning and Capability Negotiation

This specification covers versioning issues in the following areas:

- **Protocol Versions:**

  A peer that supports version 1.0 of PNM does not advertise the versions of the protocol it supports.

  This specification defines version 1.0 of PNM. The format of each message is specified in Messages (section 2.2.2) and its subsections. No additional versions are defined.

- **Capability Negotiation:**

  The protocol implicitly allows negotiation of additional capabilities by the presence or absence of **XML** elements in each message. No such capabilities are supported in version 1.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

Parameter	Value	Reference
UDP port	3702	[WS-Discovery]

2 Messages

2.1 Transport

The People Near Me (PNM) Protocol is a specialization of the Web Services Dynamic Discovery Protocol [WS-Discovery], which specifies dependent transports.

PNM messages MUST use **SOAP** version 1.2, as specified in [SOAP1.2-1/2007].

2.2 Common Message Syntax

This section contains common definitions used by this protocol. The syntax of the definitions uses **XML schema (XSD)**, as defined in [XMLSCHEMA1] and [XMLSCHEMA2], and **Web Services Description Language (WSDL)**, as defined in [WSDL].

In the XSD and examples in this document, **GUID** values are represented by the string form of **universally unique identifiers (UUIDs)**, as defined in [RFC4122] section 3.

PNM follows the message syntax of the WS-Discovery Protocol [WS-Discovery]. Additional messges are specified in Messages (section 2.2.2).

This protocol references commonly used data types as defined in [MS-DTYP].

2.2.1 Namespaces

This specification defines and references various **XML namespaces** using the mechanisms specified in [XMLNS]. Although this specification associates a specific prefix for each XML namespace that is used, the choice of any particular XML namespace prefix is implementation-specific and not significant for interoperability.

PNM specifies an XML namespace as follows:

Prefix	Namespace URI	Reference
NearMe	http://schemas.microsoft.com/p2p/2005/08/NearMe	
a	http://schemas.xmlsoap.org/ws/2004/08/addressing	[WS-Addr-Core]
d	http://schemas.xmlsoap.org/ws/2005/04/discovery	[WS-Discovery]
s	http://www.w3.org/2003/05/soap-envelope	[SOAP1.2-1/2007]

2.2.2 Messages

PNM follows the message syntax of WS-Discovery Protocol [WS-Discovery]. Messages MUST use SOAP version 1.2, as specified in [SOAP1.2-1/2007].

PNM specifies additional requirements for the following messages.

Name	Section	Description
Hello message	2.2.2.1	Used by a PNM server to announce its presence.
Bye message	2.2.2.2	Used by a PNM peer to indicate that it is disconnecting from a

Name	Section	Description
		network.
Probe message	2.2.2.3	Used by a PNM peer to solicit all the other peers on a connected **IPv6** subnet.
Probe Match message	2.2.2.4	Sent by a PNM peer after it receives a **Probe** message.

The **Resolve** and **Resolve Match** messages that are defined in [WS-Discovery] section 6 are unused and unmodified by PNM.

2.2.2.1 Hello Message

The PNM server uses the **Hello** message to announce its presence.

The format of the **Hello** message is specified in [WS-Discovery] section 4.1. The following additional constraints are placed on the </s:Envelope/s:Body/d:Hello> element:

- The <a:EndpointReference> child element MUST conform to the XML namespace format specified in Namespaces (section 2.2.1).
- A single <d:Types> child element MUST be present, and it MUST include the **NEAR_ME_TYPE** simple type specified in Simple Types (section 2.2.5).
- A single <a:Address> child element MUST be present, and it MUST contain a single instance **GUID**, as specified in Protocol Metadata (section 3.1.1.1).
- A single <NearMe:NearMeData> child element (section 2.2.3) MUST be present. This element MUST conform to the format specified in NearMeData Buffer (section 2.2.9.1).

2.2.2.2 Bye Message

A peer of PNM uses the **Bye** message to indicate that it is disconnecting from a network.

The format of the **Bye** message is specified in [WS-Discovery] section 4.1. PNM specifies no additional requirements for this message.

2.2.2.3 Probe Message

A peer of PNM uses the **Probe** message to solicit all the other peers on a connected IPv6 subnet.

The format of the **Probe** message is specified in [WS-Discovery] section 5.2. The following additional constraint is placed on the </s:Envelope/s:Body/d:Probe> element: A single <d:Types> child element MUST be present, and it MUST include the **NEAR_ME_TYPE** simple type specified in Simple Types (section 2.2.5).

2.2.2.4 Probe Match Message

A peer of PNM sends a **Probe Match** message after it receives a **Probe** message.

The format of the **Probe Match** message is specified in [WS-Discovery] section 5.3. The following additional constraints are placed on the </s:Envelope/s:Body/d:ProbeMatch> element:

- The <a:EndpointReference> child element MUST conform to the XML namespace format specified in Namespaces (section 2.2.1).

- A single <d:Types> child element MUST be present, and it MUST include the **NEAR_ME_TYPE** simple type specified in Simple Types (section 2.2.5).
- A single <a:Address> child element MUST be present, and it MUST contain a single instance GUID, as specified in Protocol Metadata (section 3.1.1.1).
- A single <NearMe:NearMeData> child element (section 2.2.3) MUST be present. This element MUST conform to the format specified in NearMeData Buffer (section 2.2.9.1).

2.2.3 Elements

The following table summarizes the set of common XML schema (XSD) element definitions defined by this specification. Element definitions that are specific to a particular operation are described with the operation.

Element	Description
<NearMe:NearMeData>	This element contains a single **base64**-encoded buffer in the format specified in NearMeData Buffer (section 2.2.9.1).

The specification of the <a:EndpointReference> element ([WS-Addr-Core] section 2.2) is constrained within messages sent by PNM, as follows:

- The <a:Address> child element MUST follow the recommended form "uuid:" followed by a globally unique identifier (GUID).
- This MUST be the instance GUID of the peer specified in Protocol Metadata (section 3.1.1.1).

2.2.4 Complex Types

This specification does not define any common XML schema (XSD) complex type definitions.

2.2.5 Simple Types

The following table summarizes the set of common XML schema (XSD) simple type definitions defined by this specification. Simple type definitions that are specific to a particular operation are described with the operation.

Simple type	Description
NEAR_ME_TYPE	This type has the value has the value **NearMe:a4c1fbe4-6d30-46c9-8bba-b8663d615706**, and it represents an instance of the peer role.

2.2.6 Attributes

This specification does not define any common XML schema (XSD) attribute definitions.

2.2.7 Groups

This specification does not define any common XML schema (XSD) group definitions.

2.2.8 Attribute Groups

This specification does not define any common XML schema (XSD) attribute group definitions.

2.2.9 Common Data Structures

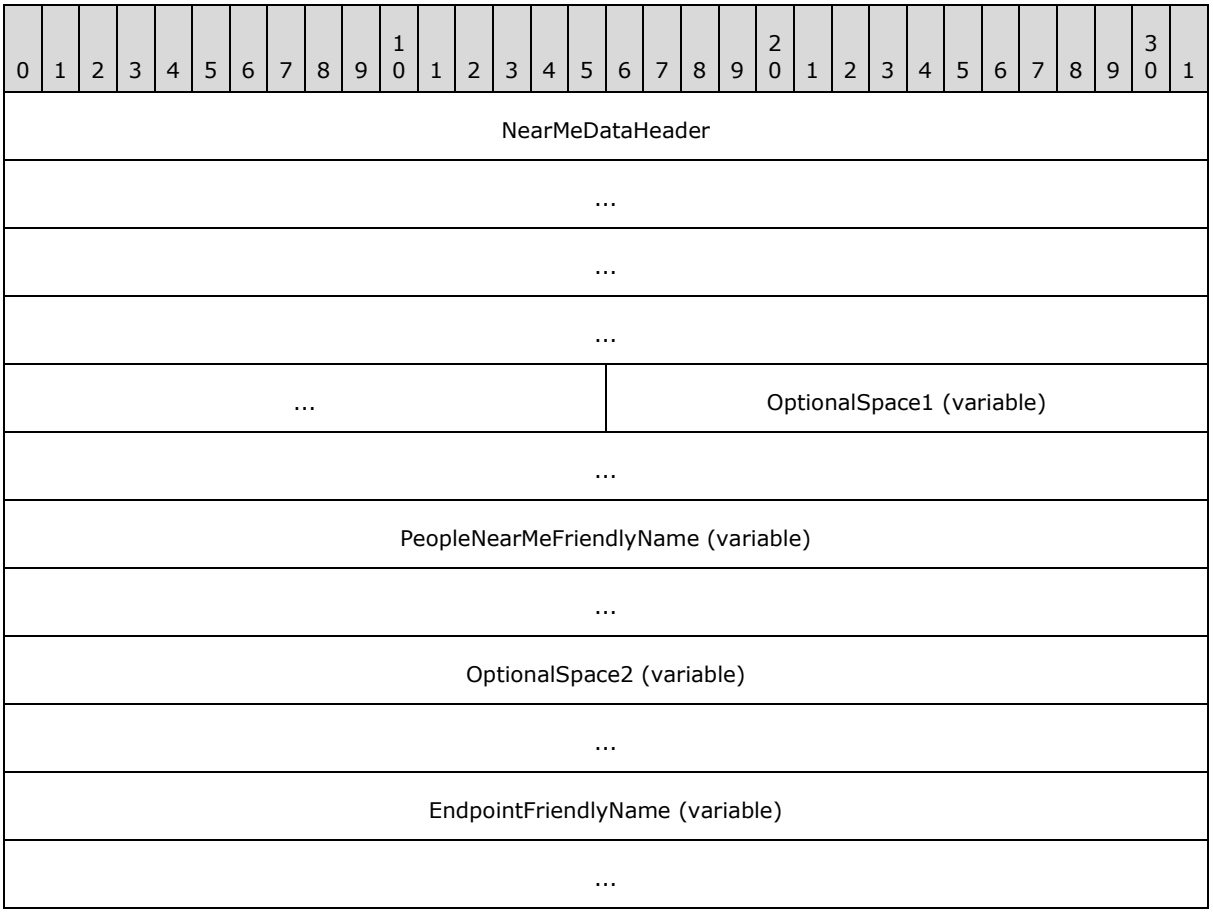
2.2.9.1 NearMeData Buffer

The **NearMeData** buffer is base64-encoded and is embedded in a <NearMe:NearMeData> child element (section 2.2.3) in the **NearMe** namespace (section 2.2.1). All multi-byte integer fields are in **host byte order**, unless explicitly specified otherwise.

2.2.9.1.1 PNM_NEARMEDATABUF

The PNM_NEARMEDATABUF structure specifies the format of the <NearMe:NearMeData> child element (section 2.2.3) in the **NearMe** namespace (section 2.2.1), which is used for PNM messages.

Note The fields that contain the **friendly name** values are not required to be contiguous with the **NearMeDataHeader** field or with each other.



NearMeDataHeader (18 bytes): A PNM_NEARMEDATAHEADER structure that specifies the locations of the fields that contain friendly name values.

OptionalSpace1 (variable): An optional, variable-length field that separates the **PeopleNearMeFriendlyName** field from the **NearMeDataHeader** field.

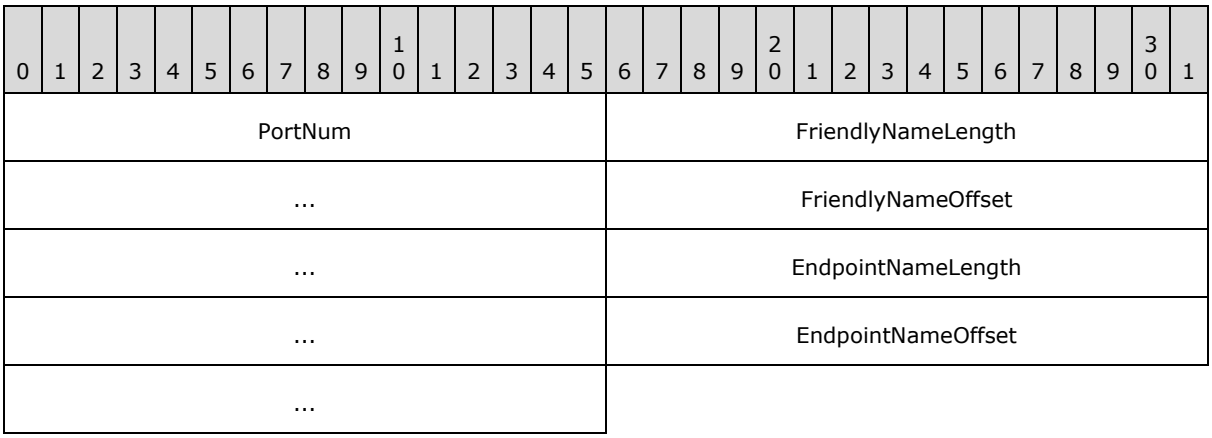
PeopleNearMeFriendlyName (variable): A **UTF-8** encoded string that specifies the **People Near Me friendly name**. Its length and location within this buffer are specified in the PNM_NEARMEDATAHEADER structure in the **NearMeDataHeader** field.

OptionalSpace2 (variable): An optional, variable-length field that separates the **EndpointFriendlyName** field from the **PeopleNearMeFriendlyName** field.

EndpointFriendlyName (variable): A UTF-8 encoded string that specifies the **endpoint friendly name**. Its length and location within this buffer are specified in the PNM_NEARMEDATAHEADER structure in the **NearMeDataHeader** field.

2.2.9.1.2 PNM_NEARMEDATAHEADER

The PNM_NEARMEDATAHEADER structure specifies the start and **endpoint references**, as well as the length of the data packet in the <NearMe:NearMeData> element (section 2.2.3) in the **NearMe** namespace (section 2.2.1), which is used for PNM messages.



PortNum (2 bytes): The **TCP** port number on which the host is listening for Peer-to-Peer Presence and Invitation Protocol [MS-P2PPI] messages. This value is represented in **network byte order**.

FriendlyNameLength (4 bytes): An unsigned integer that specifies the length, in bytes, of the People Near Me friendly name.

FriendlyNameOffset (4 bytes): An unsigned integer that specifies the offset, in bytes, from the start of the **PNM_NEARMEDATABUF** structure to the start of the People Near Me friendly name.

EndpointNameLength (4 bytes): An unsigned integer that specifies the length, in bytes, of the endpoint friendly name.

EndpointNameOffset (4 bytes): An unsigned integer that specifies the offset, in bytes, from the start of the **PNM_NEARMEDATABUF** structure to the start of the endpoint friendly name.

3 Protocol Details

3.1 Common Details

The PNM peer behaves as both a WS-Discovery target service and a WS-Discovery client, as specified in [WS-Discovery].

3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This specification does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this specification.

3.1.1.1 Protocol Metadata

A PNM peer maintains several pieces of metadata about itself:

- An unsigned 32-bit integer representing the WS-Discovery [WS-Discovery] metadata version.
- A unique GUID that is the instance GUID for WS-Discovery.<1>
- A People Near Me friendly name in **Unicode**, which the user has chosen to represent an identity in PNM.
- An endpoint friendly name in Unicode, which the user has chosen to represent an **endpoint**.

Note The preceding conceptual data can be implemented using a variety of techniques.

3.1.1.2 Peer Table

A PNM peer maintains a table of all peers discovered on every subnet for which it currently has an active IPv6 address. Each row of the table represents a single peer and includes the following data:

- PeopleNearMeFriendlyName: The peer's People Near Me friendly name in Unicode.
- EndpointFriendlyName: The peer's endpoint friendly name in Unicode.
- SourceAddress: The link-local IPv6 address from which the **Hello** or **Probe Match** message was sent.
- ID: The GUID that matches the instanceGUID in the <a:Address> child element of the **Hello** or **Probe Match** message that this peer received.
- TimeOfLastPublication: A 64-bit integer representing the system time in minutes at which the most recent **Hello** or **Probe Match** message was received from this peer. Used to determine if a peer has expired; that is, if a peer has stopped participating in PNM without sending a **Bye** message.

Note The preceding conceptual data can be implemented using a variety of techniques.

3.1.1.3 Active Subnet Table

A PNM peer maintains a table of all subnets for which it currently has an active IPv6 address. Each row of the table represents a single subnet and includes a 64-bit integer count of the number of peers discovered on that subnet.

This table is used to determine the period of the republication and expiration timers described in Timers (section 3.1.2).

Note The preceding conceptual data can be implemented using a variety of techniques.

3.1.2 Timers

Each entry in the Active Subnet Table (section 3.1.1.3) has two associated timers: a republication timer and an expiration timer.

3.1.2.1 Republication Timer

A PNM peer maintains a republication timer for each subnet for which it currently has an active IPv6 address. This timer is used to determine when to republish a **Hello** message.

The period of each republication timer is determined by the number of peers connected to the subnet, as shown in the following table.

Peers connected to the subnet	Timer period (minutes)
< 109	5
109 <= and < 516	15
<= 516 and < 1001	60
>= 1001	240

3.1.2.2 Expiration Timer

A PNM peer maintains an expiration timer for each subnet for which it currently has an active IPv6 address. This timer is used to determine when to treat a peer as expired. An expired peer is likely to have stopped participating in PNM without sending a **Bye** message.

The period of each expiration timer is determined by the number of peers connected to the subnet, as shown in the following table.

Peers connected to the subnet	Timer period (minutes)
< 109	5
109 <= and < 516	15
<= 516 and < 1001	60
>= 1001	240

3.1.3 Initialization

When PNM is initialized, the PNM peer performs the following actions:

- The peer MUST set its metadata version to one.
- The peer MUST create a unique instance GUID.
- The peer MUST send a Hello Message (section 2.2.2.1).
- The peer SHOULD send a Probe Message (section 2.2.2.3). <2>
- The peer MUST begin listening for messages. Transport information is defined in [WS-Discovery] section 2.4.

3.1.4 Message Processing Events and Sequencing Rules

A PNM peer MUST verify that each received message matches the schema in [WS-Discovery] Appendix III and discards malformed messages. Further parsing depends on the message type.

PNM specifies additional processing for the following messages.

Name	Section	Description
Hello message	3.1.4.1	Used by a PNM server to announce its presence.
Bye message	3.1.4.2	Used by a PNM peer to indicate that it is disconnecting from a network.
Probe message	3.1.4.3	Used by a PNM peer to solicit all the other peers on a connected IPv6 subnet.
Probe Match message	3.1.4.4	Sent by a PNM peer after it receives a **Probe** message.
Recovery message	3.1.4.5	Used to retrieve network transport information for a target service if it has an endpoint reference, as specified in [WS-Discovery].

All other messages MUST be discarded without further processing.

3.1.4.1 Hello Message

An implementation of PNM MUST observe the requirements specified in [WS-Discovery] section 4.1 when sending and receiving **Hello** messages. PNM adds the following additional requirements:

- When a peer receives a **Hello** message, it MUST verify that the message satisfies the requirements of Namespaces (section 2.2.1) and the Hello Message (section 2.2.2.1) of this specification.
- The peer MUST verify that the message originated from a link-local IPv6 address, not a site-local or global IPv6 address.
- The peer MUST parse the <NearMe:NearMeData> child element (section 2.2.3) of the message and verify that it satisfies the structure requirements of the NearMeData Buffer (section 2.2.9.1).
- If the **Hello** message does not meet these requirements, it MUST be silently discarded.

The peer MUST determine if there is already an entry in the Peer Table (section 3.1.1.2) for the data in the <NearMe:NearMeData> child element. If an entry is present, the peer MUST set the **TimeOfLastPublication** field of the entry to the current time, and processing is complete.

If an entry in the Peer Table does not already exist, processing of the message continues as follows:

- A new entry MUST be added to the Peer Table.
- The **TimeOfLastPublication** field of the entry MUST be set to the current time.
- The **NumPeers** field MUST be incremented in the row corresponding to the subnet on which the **Hello** message was received, in the Active Subnet Table (section 3.1.1.3).
- The period of the republication timer MUST be computed according to the formula described in Republication Timer (section 3.1.2.1), and the timer MUST be adjusted if necessary.
- The period of the expiration timer MUST be computed according to the formula described in Expiration Timer (section 3.1.2.2), and the timer MUST be adjusted if necessary.

3.1.4.2 Bye Message

An implementation of PNM MUST observe the requirements specified in [WS-Discovery] section 4.2 when sending and receiving **Bye** messages. PNM adds the additional requirement that when the peer receives a **Bye** message, it MUST verify that the </s:Body/d:Bye/a:EndpointReference/a:Address> element contains the instance GUID. If not, the message MUST be silently discarded.

If the instance GUID is not the null GUID, which is entirely filled with zeroes, the peer MUST remove the entry from the Peer Table (section 3.1.1.2) whose **ID** field matches the instance GUID in the **Bye** message. The peer MUST also decrement the NumPeers field in the entry that corresponds to the subnet on which the **Bye** message was received, in the Active Subnet Table (section 3.1.1.3).

3.1.4.3 Probe Message

An implementation of the PNM MUST observe the requirements specified in [WS-Discovery] section 5.2 when sending and receiving **Probe** messages. PNM adds the additional requirement that when a peer receives a **Probe** message, it MUST verify that the message satisfies the requirements of Namespaces (section 2.2.1) and the Probe Message (section 3) of this specification. If not, the message MUST be silently discarded.

When a matching **Probe** message is received, the peer MUST reply with a **Probe Match** message, following the rules in [WS-Discovery] section 5.3 and in Probe Match Message (section 2.2.2.4).

3.1.4.4 Probe Match Message

An implementation of PNM MUST observe the requirements specified in [WS-Discovery] section 5.3 when sending and receiving **Probe Match** messages. PNM adds the following additional requirements:

- When a peer receives a **Probe Match** message, it MUST verify that the message satisfies the requirements in Namespaces (section 2.2.1) and Probe Match Message (section 2.2.2.4) of this document.
- The peer MUST verify that the message originated from a link-local IPv6 address, not a site-local or global IPv6 address.
- The peer MUST parse the <NearMe:NearMeData> child element (section 2.2.3) of the message and verify that it satisfies the structure requirements of the NearMeData Buffer (section 2.2.9.1).
- If the **Probe Match** message does not meet these requirements, it MUST be silently discarded.

If there is already an entry in the Peer Table (section 3.1.1.2) for the data in the <NearMe:NearMeData> child element, the peer MUST set the **TimeOfLastPublication** field of the entry to the current time, and processing is complete.

If an entry in the Peer Table does not already exist, processing of the message continues as follows:

- A new entry MUST be added to the Peer Table.
- The **TimeOfLastPublication** field of the entry MUST be set to the current time.
- The **NumPeers** field MUST be incremented in the row corresponding to the subnet on which the **Probe Match** message was received, in the Active Subnet Table (section 3.1.1.3).
- The period of the republication timer MUST be computed according to the formula described in Republication Timer (section 3.1.2.1), and the timer MUST be adjusted if necessary.
- The period of the expiration timer MUST be computed according to the formula described in Expiration Timer (section 3.1.2.2), and the timer MUST be adjusted if necessary.

3.1.4.5 Resolve Message

The **Resolve** and **Resolve Match** messages that are defined in [WS-Discovery] section 6 are unmodified by PNM.

[WS-Discovery] section 6.1 specifies that a server responds to a **Resolve** message that matches its endpoint reference. PNM relaxes this requirement.

When the peer receives a **Resolve** message, it **MAY** ignore the message. <3> If the server chooses to process **Resolve** messages, then it MUST follow the rules in [WS-Discovery] section 6.2.

In [WS-Discovery], the **Resolve**/**Resolve Match** message exchange is used when a client has the endpoint reference of a server but requires its **transport addresses (XAddrs)**. PNM uses only endpoint references, so obtaining transport addresses (XAddrs) is not necessary.

3.1.5 Timer Events

An implementation of PNM MUST implement the timers defined in [WS-Discovery] sections 2.4, 3, and 7. PNM defines the following additional timers:

3.1.5.1 Republication Timer

PNM requires periodic republication of **Hello** messages to determine if a peer is still online. This is intended to avoid stale entries in the Peer Table (section 3.1.1.2). Stale entries could be created when a peer goes offline without publishing a **Bye** message.

The period of a republication timer is determined by the formula specified in section 3.1.2.1 of this specification. The timer is first set when the first **Hello** message is sent. When a republication timer fires, a peer MUST send a **Hello** message on all multicast-capable IPv6 addresses.

3.1.5.2 Expiration Timer

The period of an expiration timer is determined by the formula specified in section 3.1.2.2 of this specification. When an expiration timer fires, a peer MUST remove all expired entries from the Peer Table (section 3.1.1.2) and decrement the NumPeers field in the entry corresponding to the subnet on which the **expired** peer was discovered, in the Active Subnet Table (section 3.1.1.3).

3.1.6 Other Local Events

3.1.6.1 Attach to a Subnet

When a peer host receives a link-local IPv6 address on a subnet which supports multicast, the peer MUST check whether any existing row in the Active Subnet Table (section 3.1.1.3) matches the subnet on which the **Hello** message was received. If no row matches, then the client MUST process the event as follows:

- Create a new row containing a single entry.
- Set the entry's NumPeers field to 0.
- Send a **Hello** message multicast on all network interfaces that support multicast and have a valid link-local IPv6 address.
- Send a **Probe** message that is multicast on all network interfaces that support multicast and have a valid link-local IPv6 address.

3.1.6.2 Detach from a Subnet

When a peer host detaches from an IP subnet, the peer MUST find the row with the matching subnet in the Active Subnet Table (section 3.1.1.3). The peer MUST then process the event as follows:

- Delete the row.
- Send a **Hello** message multicast on all network interfaces that support multicast and have a valid link-local IPv6 address.
- Send a **Probe** message that is multicast on all network interfaces that support multicast and have a valid link-local IPv6 address.

3.1.6.3 Shut Down

When PNM is shut down, the peer performs the following actions:

- The peer MUST ignore further incoming messages.
- The peer MUST send a Bye Message (section 2.2.2.1).
- The peer SHOULD close the network ports defined in [WS-Discovery] section 2.4.<4>

3.2 Server Details

This specification does not define any server specific details.

3.3 Client Details

This specification does not define any client specific details.

4 Protocol Examples

4.1 A Hello Message When a Peer Starts

In this example, a PNM peer with a People Near Me friendly name, "eliotf", and endpoint friendly name, "EF-64", resides on a host that is connected to a single network, holding a single IPv6 address. When the PNM peer is started, the host sends the following message.

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope
  xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:wsd="http://schemas.xmlsoap.org/ws/2005/04/discovery"
  xmlns:NearMe="http://schemas.microsoft.com/p2p/2005/08/NearMe">
  <soap:Header>
    <wsa:To>
      urn:schemas-xmlsoap-org:ws:2005:04:discovery
    </wsa:To>
    <wsa:Action>
      http://schemas.xmlsoap.org/ws/2005/04/discovery/Hello
    </wsa:Action>
    <wsa:MessageID>
      urn:uuid:16d1ca53-23c0-4e27-accf-2bf71377f49e
    </wsa:MessageID>
    <wsd:AppSequence
      InstanceId="0" MessageNumber="1" >
    </wsd:AppSequence>
  </soap:Header>
  <soap:Body>
    <wsd:Hello>
      <wsa:EndpointReference>
        <wsa:Address>
          uuid:A99558EB-C1D8-49D3-9476-8B9A6571800B
        </wsa:Address>
      </wsa:EndpointReference>
      <wsd:Types>
        NearMe:a4c1fbe4-6d30-46c9-8bba-b8663d615706
      </wsd:Types>
      <wsd:MetadataVersion>
        1
      </wsd:MetadataVersion>
      <NearMe:NearMeData>
        0M4AAAgAAAAUAAAABwAAABwAAABlbGlvdGYAAEVGLTY0AAA=
      </NearMe:NearMeData>
    </wsd:Hello>
  </soap:Body>
</soap:Envelope>
```

The packet is sent twice, as specified in [WS-Discovery], to the IPv6 address FF02::C port 3702, which is the standard WS-Discovery IPv6 multicast endpoint.

When the PNM peer is shut down, the following **Bye** message is sent. Like the **Hello** message, it is sent twice to the IPv6 multicast address.

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope
  xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:wsd="http://schemas.xmlsoap.org/ws/2005/04/discovery"
  xmlns:NearMe="http://schemas.microsoft.com/p2p/2005/08/NearMe">
  <soap:Header>
    <wsa:To>
      urn:schemas-xmlsoap-org:ws:2005:04:discovery
    </wsa:To>
    <wsa:Action>
      http://schemas.xmlsoap.org/ws/2005/04/discovery/Bye
    </wsa:Action>
    <wsa:MessageID>
      urn:uuid:e1c429f4-661d-4f98-a6d3-ce712efa28b7
    </wsa:MessageID>
    <wsd:AppSequence
      InstanceId="0"
      MessageNumber="2" >
    </wsd:AppSequence>
  </soap:Header>
  <soap:Body>
    <wsd:Bye>
      <wsa:EndpointReference>
        <wsa:Address>
          uuid:A99558EB-C1D8-49D3-9476-8B9A6571800B
        </wsa:Address>
      </wsa:EndpointReference>
    </wsd:Bye>
  </soap:Body>
</soap:Envelope>
```

4.2 A Peer Probe Message with Probe Match Replies

In this example, a PNM peer with People Near Me friendly name, "other-guy", and endpoint friendly name, "Other Computer", resides on a host that is connected to a single network, holding a single IPv6 address.

When the PNM peer initiates a discovery, the host sends the following message. Line 20 indicates that the search is for PNM peers.

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope
  xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:wsd="http://schemas.xmlsoap.org/ws/2005/04/discovery" >
  <soap:Header>
    <wsa:To>
      urn:schemas-xmlsoap-org:ws:2005:04:discovery
    </wsa:To>
    <wsa:Action>
      http://schemas.xmlsoap.org/ws/2005/04/discovery/Probe
    </wsa:Action>
    <wsa:MessageID>
      urn:uuid:7895122d-f9d6-4cb9-b819-872f24c271b9
    </wsa:MessageID>
```

```
  </soap:Header>
  <soap:Body>
    <wsd:Probe>
      <wsd:Types>
        NearMe:a4c1fbe4-6d30-46c9-8bba-b8663d615706
      </wsd:Types>
    </wsd:Probe>
  </soap:Body>
</soap:Envelope>
```

The message is sent twice to the IPv6 address FF02::C port 3702. A PNM peer on the same subnet receives the probe.

The receiving peer has the People Near Me friendly name, "eliotf", and the endpoint friendly name, "EF-64". After 150 milliseconds it replies with the following message:

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope
  xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:wsd="http://schemas.xmlsoap.org/ws/2005/04/discovery"
  xmlns:NearMe="http://schemas.microsoft.com/p2p/2005/08/NearMe">
  <soap:Header>
    <wsa:To>
      http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
    </wsa:To>
    <wsa:Action>
      http://schemas.xmlsoap.org/ws/2005/04/discovery/ProbeMatches
    </wsa:Action>
    <wsa:MessageID>
      urn:uuid:769dfff7-e778-4506-a764-0cbb26cb0f60
    </wsa:MessageID>
    <wsa:RelatesTo>
      urn:uuid:7895122d-f9d6-4cb9-b819-872f24c271b9
    </wsa:RelatesTo>
    <wsd:AppSequence
      InstanceId="0"
      MessageNumber="4">
    </wsd:AppSequence>
  </soap:Header>
  <soap:Body>
    <wsd:ProbeMatches>
      <wsd:ProbeMatch>
        <wsa:EndpointReference>
          <wsa:Address>
            uuid:FDEFC35B-3B18-4E1C-B970-09F811D00304
          </wsa:Address>
        </wsa:EndpointReference>
        <wsd:Types>
          NearMe:a4c1fbe4-6d30-46c9-8bba-b8663d615706
        </wsd:Types>
        <wsd:MetadataVersion>
          1
        </wsd:MetadataVersion>
      </wsd:ProbeMatch>
    </wsd:ProbeMatches>
```

```
      <NearMe:NearMeData>
        0M4AAAgAAAAUAAAABwAAABwAAABlbGlvdGYAAEVGLTY0AAA=
      </NearMe:NearMeData>
    </soap:Body>
  </soap:Envelope>
```

The UUID in line 42 matches the one in line 45 of the **Probe** message. Line 66 contains the base64 encoded data that is specified in NearMeData Buffer (section 2.2.9.1).

5 Security

5.1 Security Considerations for Implementers

5.1.1 Potential for High Unicast Traffic

Neither [WS-Discovery] nor the People Near Me (PNM) Protocol provides a way to control the number or pace of replies to a **Probe** message. In a very large network, a client may be overwhelmed by many server replies.

5.1.2 Lack of Message Authentication

PNM does not provide an authentication method for messages. The possibility exists for a malicious host to send incorrect Hello (section 2.2.2.1), Bye (section 2.2.2.2), and Probe Match (section 2.2.2.4) messages in order to confuse a peer.

5.2 Index of Security Parameters

None.

6 Appendix A: Full WSDL

The WSDL description for the People Near Me (PNM) Protocol is provided in the Web Services Dynamic Discovery Protocol [WS-Discovery].

7 Appendix B: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- Windows Vista operating system
- Windows 7 operating system

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

<1> Section 3.1.1.1: Windows uses this GUID to fill the **id** member of the **PEER_PEOPLE_NEAR_ME** structure that the Peer to Peer Presence and Invitation Protocol [MS-P2PPI] uses to represent a "Person Near Me". It is generated the first time a user signs in to PNM. For more information, see [MSDN-P2PC].

<2> Section 3.1.3: Windows sends a **Probe** message.

<3> Section 3.1.4.5: Windows ignores **Resolve** messages.

<4> Section 3.1.6.3: Windows closes the ports if and only if no other [WS-Discovery] based protocol is active on the host.

8 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

9 Index

T

V