[MS-PSDP]:
Proximity Service Discovery Protocol Specification

Revision Summary

Date	Revision History	Revision Class	Comments
02/22/2007	0.01		MCPP Milestone 3 Initial Availability
06/01/2007	1.0	Major	Updated and revised the technical content.
07/03/2007	1.0.1	Editorial	Revised and edited the technical content.
07/20/2007	1.0.2	Editorial	Revised and edited the technical content.
08/10/2007	1.0.3	Editorial	Revised and edited the technical content.

Date	Revision History	Revision Class	Comments
09/28/2007	1.0.4	Editorial	Revised and edited the technical content.
10/23/2007	1.0.5	Editorial	Revised and edited the technical content.
11/30/2007	1.0.6	Editorial	Revised and edited the technical content.
01/25/2008	1.0.7	Editorial	Revised and edited the technical content.

Table of Contents

1 Introduction

This specification defines a Microsoft proprietary protocol that is referred to as the Proximity Service Discovery Protocol. The Proximity Service Discovery Protocol allows a client to discover services in its physical proximity, which is defined by the radio range.

1.1 Glossary

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

Access Point (AP)
Hash Function
Keyed Hash Message Authentication Code (HMAC)
MAC Sublayer Management Entity (MLME)
Station (STA)
Station Management Entity (SME)
Unicode
Uniform Resource Identifier (URI)

The following terms are specific to this document:

Ad Hoc Network: A self-configuring wireless network of mobile routers (and associated hosts) that are connected by wireless links, the union of which form an arbitrary topology. See [IEEE802.11].

Basic Service Set (BSS): "A set of **stations** controlled by a single coordination function," as defined in [IEEE802.11] section 3.7.

Hash: A term that refers to either a **hash function**, the value computed by such a function, or the act of computing such a value.

Independent Basic Service Set (IBSS): A **basic service set (BSS)** that is an autonomous network, as defined in [IEEE802.11] section 3.27. An **IBSS** does not provide access to a distribution system.

Information Element: Within 802.11 management frames, information elements are fields used to encode variable-length mandatory and all optional body components, as defined in [IEEE802.11] section 7.3.

Medium Access Control (MAC): A data communication protocol sublayer that is part of the seven-layer OSI model data-link layer (layer 2). It provides addressing and channel access control mechanisms that make it possible for several terminals or network nodes to communicate within a multipoint network, typically a **local area network (LAN)**.

Medium Access Control (MAC) protocol data unit (MPDU): The unit of data exchanged between two peer **MAC** entities using the services of the physical layer.

Namespace: An abstract container that provides context for the items (names, technical terms, or words) that it holds and allows disambiguation of items that have the same name (residing in different **namespaces**).

Octet: A group of 8 bits often referred to as a byte.

Organizationally Unique Identifier (OUI): A unique 24-bit string that is assigned to computer hardware manufacturers, as specified in [IEEE OUI].

Secure Hash Algorithm (SHA): A **hash function** that refers to any of the five Federal Information Processing Standards Publications (FIPS PUBS)–approved algorithms for computing a condensed digital representation (known as a message digest) that is, to a high degree of probability, unique for a specified input data sequence (the message). For more information, see [SHA256].

Service Access Point (SAP): An identifying label for network endpoints that are used in Open Systems Interconnection (OSI) networking. The **SAP** is a conceptual location at which one OSI layer can request the services of another OSI layer.

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

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.

[IEEE802.11] Institute of Electrical and Electronics Engineers, "IEEE Standards for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Network - Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", ANSI/IEEE Std 802.11, 1999, http://standards.ieee.org/getieee802/download/802.11-1999.pdf

[IEEE802.11-2007] Institute of Electrical and Electronics Engineers, "Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", 11-2007, http://ieeexplore.ieee.org/servlet/opac?punumber=4248376

Note There is a charge to download this document.

[IEEE OUI] IEEE Standards Association, "IEEE OUI Registration Authority", February 2007, http://standards.ieee.org/regauth/oui/oui.txt

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary", March 2007.

[RFC2104] Krawczyk, H., Bellare, M., and Canetti, R., "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997, http://www.ietf.org/rfc/rfc2104.txt

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

[RFC3986] Berners-Lee, T., Fielding, R., and Masinter, L., "Uniform Resource Identifier (URI): Generic Syntax", RFC 3986, January 2005, http://www.ietf.org/rfc/rfc3986.txt

[RFC4634] Eastlake III, D. and Hansen, T., "US Secure Hash Algorithms (SHA and HMAC-SHA)", RFC 4634, July 2006, http://www.ietf.org/rfc/rfc4634.txt

[SHA256] National Institute of Standards and Technology, "FIPS 180-2, Secure Hash Standard (SHS)", August 2002, http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf

1.2.2 Informative References

[RFC2608] Guttman, E., Perkins, C., Veizades, J., and Day, M., "Service Location Protocol, Version 2", RFC 2608, June 1999, http://www.ietf.org/rfc/rfc2608.txt

[UPNPARCH1] UPnP Forum, "UPnP Device Architecture 1.0", July 2006, http://www.upnp.org/specs/arch/UPnP-DeviceArchitecture-v1.0-20060720.pdf

[WS-Discovery] Beatty, J. et. al, "Web Services Dynamic Discovery (WS-Discovery)", April 2005, http://www1.webmethods.com/PDF/WS-Discovery.pdf

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

1.3 Protocol Overview

The purpose of the Proximity Service Discovery Protocol is to convey service discovery information, such as service advertisements, as part of Beacon frames, as specified in [IEEE802.11]. Beacon frames are periodically broadcast by **access points (APs)** and **stations (STAs)**, as specified in [IEEE802.11], that operate in ad hoc mode. According to the IEEE802.11 protocol, stations in radio range, that is, in proximity, receive and process Beacon frames during a normal channel scan operation.

The Beacon frame can contain single or multiple proprietary **information elements** that carry discovery information pertaining to the services that the device offers. Proprietary information elements are identified by their Element IDs and are further distinguished by an IEEE-administered **Organizationally Unique Identifier (OUI)** and a predefined OUI Type value.

A format identifier describes the format of the information carried in the information element. To ensure uniqueness while circumventing the need for central administration of format identifiers, a string in the form of a **Uniform Resource Identifier (URI)**, as specified in [RFC3986], could be used to distinguish the format. However, because the transmission must be efficient and space in the information element is limited, the string is not actually transmitted, but, instead, its **hash** is transmitted. On the client, which is the receiving side of the beacon, the hash is matched against a known set of format identifiers.

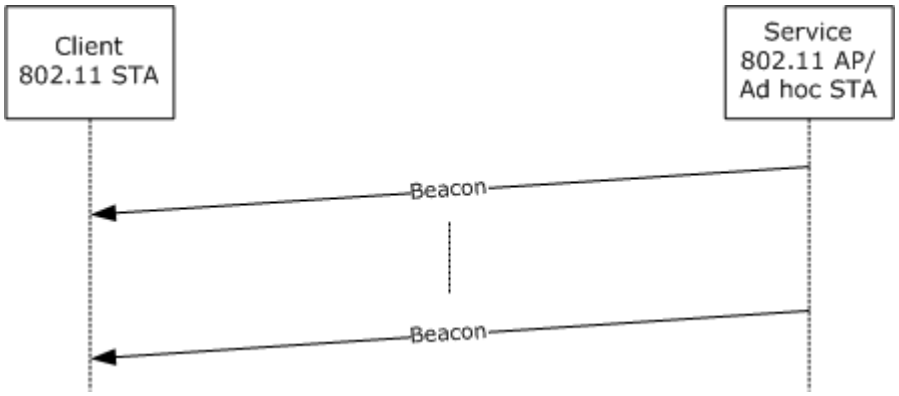


Figure 1: PSDP client/server communication

The preceding diagram illustrates the relationship between a service-bearing device, which is an AP or ad hoc station, as specified in [IEEE802.11], and the client that acts as a station, as specified in [IEEE802.11], in ad hoc or infrastructure mode.

A client that is in discovery mode (that is, searching for a service in its physical proximity) picks up the beacon during its regular scan intervals. It processes the beacon for known discovery information elements based on the OUI and OUI Type,<1> and it notifies the application if the format identifier that is represented by the hash matches any known format identifiers. Data carried in the information element is opaque to the protocol.<2> It is the responsibility of the application to resolve possible hash collisions. The application can do so by examining the data carried in the information element or by re-issuing a discovery request at a higher layer by using the full format identifier string after a connection has been established.

The inclusion of service discovery information in broadcast messages enables the discovery of services before connecting to the service-hosting device.<3>

1.4 Relationship to Other Protocols

The Proximity Service Discovery Protocol extends the IEEE802.11 standard, whose conventions are applied as specified in [IEEE802.11]. The Proximity Service Discovery Protocol introduces a specific use for one of that protocol's reserved information element types, and it defines additional **MAC** layer abstract service primitives for managing the configuration, transmission, and receipt of these new information elements.

1.5 Prerequisites or Preconditions

In the Proximity Service Discovery Protocol, the service-hosting device acts as an AP or ad hoc station and includes an additional information element (discovery information element) in its periodically transmitted beacon. The client acts as a station in infrastructure or ad hoc mode and is able to extract the discovery information element, as specified in section 2.2, from the received beacon.

1.6 Applicability Statement

The Proximity Service Discovery Protocol works with higher-layer discovery protocols, such as the Simple Service Discovery Protocol, as specified in [UPNPARCH1], Web Services Dynamic Discovery (WS-Discovery), as specified in [WS-Discovery], and the Service Location Protocol, Version 2, as specified in [RFC2608]. The Proximity Service Discovery Protocol facilitates discovery before connecting on a wireless medium.

The discovery advertisements of these related protocols can be mapped into discovery information elements that are conveyed in IEEE802.11 beacons. A unique format identifier can be defined for each higher-layer protocol based on the Uniform Resource Identifier (URI) **namespace** of the respective higher-layer discovery protocol.

1.7 Versioning and Capability Negotiation

This protocol does not define any capability negotiation or versioning aspects.

1.8 Vendor-Extensible Fields

Vendors can use any combination of data for the content of the discovery information element. However, vendors SHOULD define a valid URI to identify a proprietary format. Vendors SHOULD NOT use URIs that represent well-known namespaces when they devise proprietary formats.

1.9 Standards Assignments

Parameter	Value	Reference
Vendor-specific Element ID	221	As specified in [IEEE P802.11]
OUI	00-50-f2	As specified in [IEEE OUI]

2 Messages

The following sections specify how Proximity Service Discovery Protocol messages are transported and also specify Proximity Service Discovery Protocol message syntax.

2.1 Transport

Single or multiple discovery information elements are transmitted as part of IEEE802.11 Beacon frames. There are no requirements for the order of the information elements. However, the size of individual information elements MUST NOT exceed 255 **octets**, including the Element ID, the **Length** field, the Organizationally Unique Identifier (OUI), the OUI Type, the format identifier hash, and the discovery data.

The format of information elements is specified in [IEEE802.11] section 7.3.2. The format and processing of Beacon and Probe Response frames are also specified in [IEEE802.11].

To improve transmission behavior and to reduce processing and hardware requirements, the total size of the discovery information element data SHOULD be kept small.

If multiple information elements are included in the Beacon or Probe Response frame, the maximum size of the Beacon frame, as specified in [IEEE802.11], MUST NOT be exceeded.

Duplicate information elements MAY be discarded by the receiver.<4> However, Windows does not discard duplicate information elements.

Because of the characteristics of the wireless medium, there is no guarantee that a Beacon frame will be successfully received by any other station. Therefore, a station that is operating under the rules of the **independent basic service set (IBSS)** or an AP SHOULD include the information element in multiple successive beacons.

If the discovery information element was transmitted in the beacon, an IEEE802.11 AP or IEEE802.11 ad hoc station (STA) SHOULD also include the discovery information element in the Probe Response frame upon reception of a Probe frame. The rules for sending a Probe response are unchanged from the rules as specified in [IEEE802.11] section 11.1.3.2.1.

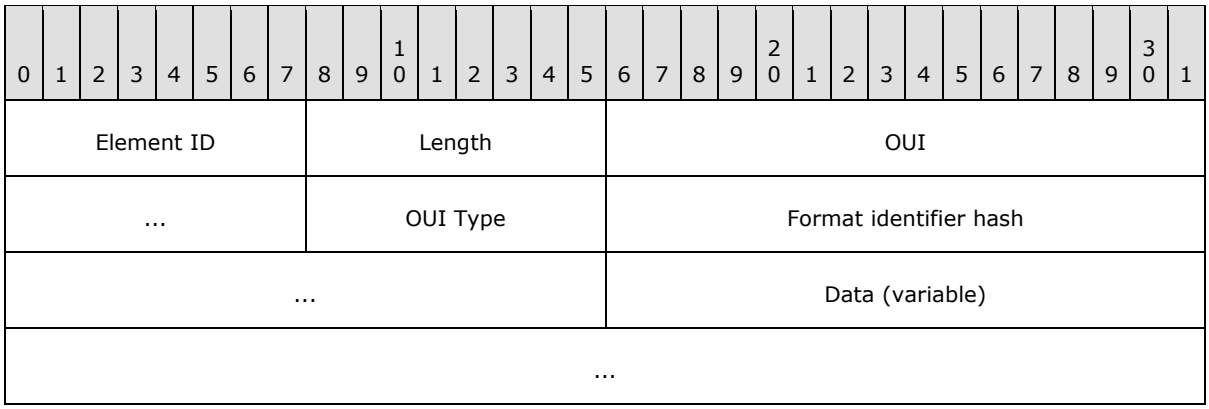
If a service is advertised by a station that operates under the rules of the IBSS, other stations that are members of the same IBSS MUST NOT replicate the discovery information elements when they transmit the beacon.

2.2 Message Syntax

The following sections specify Proximity Service Discovery Protocol message syntax.

2.2.1 Structure of the Discovery Information Element

The structure of the discovery information element is shown in the following packet.



Element ID (1 byte): Contains the ID of the element as specified in [IEEE802.11-2007] section 7.3.2. It MUST contain a value of 221, identifying a vendor-specific element (as specified in [IEEE802.11-2007] table 26) in which the vendor is identified by an IEEE-issued OUI in a subsequent field.

Length (1 byte): Contains the length of the **Data** field in octets plus 8.

OUI (3 bytes): The IEEE-assigned OUI for Microsoft. The **OUI** field MUST contain a value of (00:50:f2) as specified in [IEEE OUI].

OUI Type (1 byte): A packet subtype within the universe specific to a particular OUI value. For the Proximity Service Discovery Protocol, the OUI Type MUST contain a value of 6.

Format identifier hash (4 bytes): A value that identifies the format of the data, as specified in section 2.2.2.

Data (variable): Contains user-defined data for discovery.

The transmission order follows the conventions for transmission of **MAC protocol data units (MPDUs)**, as specified in [IEEE802.11] section 7.

The message format of the Beacon frame is as specified in [IEEE802.11] section 7.2.3.1. The message format of the Probe Response frame is as specified in [IEEE802.11] section 7.2.3.9.

2.2.2 Calculation of the Format Identifier Hash

The **format identifier hash** is represented by the bits 0…31 of **keyed-hash message authentication code (HMAC)** (as specified in [RFC4634]) based on the SHA-256 algorithm, as specified in [SHA256], over the format identifier string, as specified in [RFC4634].

Note Sample code for the calculation of the HMAC is as specified in [RFC4634].

The key used is the "null" key (null pointer to the authentication key, and zero length authentication key per the source code in [RFC4634]).

The characters and spaces of the format identifier string are encoded as **Unicode** UTF-16 using little-endian representation.

The hash of the format identifier is transmitted beginning with the first octet of the octet array.

For sample format identifier strings and their corresponding hashes, see section 4.

3 Protocol Details

The following sections specify details of the Proximity Service Discovery Protocol, including semantics of service primitives for the client and server.

3.1 Layer Management

Protocol primitives are exchanged as layer management primitives by means of the MLME_SAP, which includes the **MAC sublayer management entity (MLME)** and **service access points (SAPs)**.

Definitions of primitives and terms are as specified in [IEEE802.11] section 10. This layer management model is identical to the model that is specified in [IEEE802.11] section 10.1. The following figure illustrates the layer model and represents a subset of Figure 63, as specified in [IEEE802.11] section 10.2.

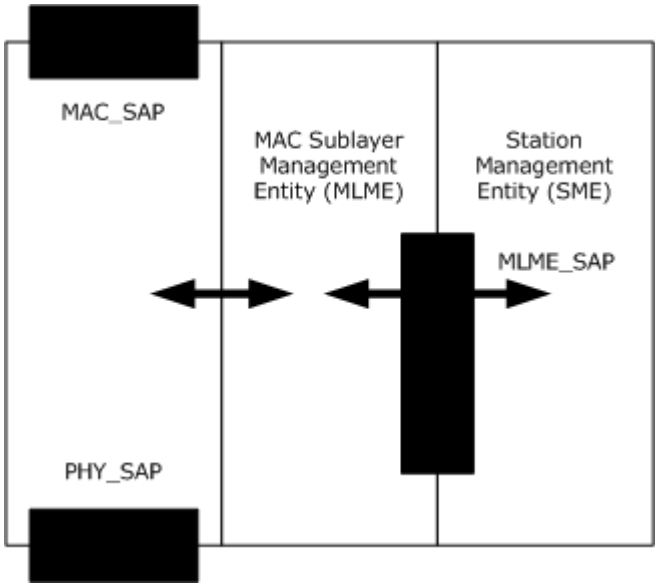


Figure 2: Proximity Service Discovery Protocol (PSDP) dependencies

The Server Details (section 3.2) and Client Details (section 3.3) sections of this document describe the MLME primitives that are exchanged with the **station management entity (SME)** in order for the server and client to configure and receive discovery information elements.

3.2 Server Details

The server SHOULD transmit the discovery information elements as part of the IEEE802.11 beacon. Therefore, the server function MUST be hosted by either an IEEE802.11 AP or an ad hoc station.

The beacon MAY contain single or multiple information elements.<5>

To compensate for an unreliable transmission over the wireless medium, the information elements SHOULD be contained in multiple successive Beacon frames.

3.2.1 MLME-PSD-Transmit.request

3.2.1.1 Semantics of the Service Primitive

The contents of a discovery information element and its format identifier are passed to the MLME SAP by using the MLME-PSD-Transmit.request primitive.

```
MLME-PSD-Transmit.request
(Discovery data,
FormatIdentifier)
```

3.2.1.2 When the Primitive Is Generated

The primitive is generated by the SME in order to start transmitting the discovery information element as part of any Beacon and Probe Responses that are subsequently transmitted. A request that does not contain any discovery data cancels the transmission of a previously registered discovery information element for the same format identifier.

3.2.1.3 Effect on Receipt

Upon receipt of the primitive, a proprietary discovery information element, as specified in section 2.2, is created. The hash of the format identifier, in addition to the discovery data, is included in the information element. The information element MUST be transmitted in all IEEE802.11 Beacon frames following the request and SHOULD also be included in IEEE802.11 Probe Responses. The MLME subsequently issues an MLME-PSD-Transmit.confirm that reflects the result of the MLME-PSD.Transmit.request.

3.2.2 MLME-PSD-Transmit.confirm

3.2.2.1 Semantics of the Service Primitive

The primitive parameters are as follows:

```
MLME-PSD-Transmit.confirm
(ResultCode)
```

The ResultCode indicates the success of the MLME-PSD-Transmit.request and is an enumeration of the following:

- SUCCESS
- Invalid Parameters
- No resources

3.2.2.2 When the Primitive Is Generated

The primitive is generated by the MLME to indicate the commencement of the Beacon and Probe Response transmission containing the information element. It is not generated until the information element is configured.

3.2.2.3 Effect on Receipt

The SME is notified that the discovery information element will be included in subsequent Beacon and Probe Response transmissions.

3.3 Client Details

The client SHOULD regularly scan its radio spectrum for beacons. When a beacon is received, the client SHOULD examine it for the presence of discovery information elements, and if any are present, compare their format identifier hashes with the known hashes of format identifiers that were previously registered (via MLME-PSD-Config.request). If the client finds a match, it SHOULD notify the client discovery application.

3.3.1 MLME-PSD-Config.request

3.3.1.1 Semantics of the Service Primitive

The format identifier is passed to the MLME SAP by using the following primitive:

```
MLME-PSD-Config.request
(FormatIdentifier)
```

3.3.1.2 When the Primitive Is Generated

The primitive is generated by the SME to register the format identifier of a discovery information element that should be indicated to the SME. Multiple MLME-PSD-Config.requests MAY be sent to register multiple format identifiers.

3.3.1.3 Effect on Receipt

Upon receipt of the primitive, an MLME-PSD-Config.confirm is generated and sent to the SME. After registration of a format identifier, subsequently received Beacon and Probe Response frames are examined for discovery information elements and matched against any format identifiers that have been registered. Discovery information elements are passed to the SME by means of the MLME-PSD-Receive.indication.

3.3.2 MLME-PSD-Config.confirm

3.3.2.1 Semantics of the Service Primitive

The primitive parameters are as follows:

```
MLME-PSD-Config.confirm
(ResultCode)
```

The ResultCode indicates the success of the MLME-PSD-Config.request and is an enumeration of the following:

- SUCCESS
- Invalid Parameters

- NOT Supported
- No resources

3.3.2.2 When the Primitive Is Generated

The primitive is generated by the MLME upon completion of an MLME-PSD-Config.request operation to indicate the success or failure of format identifier registration. It is not generated until the operation completes.

3.3.2.3 Effect on Receipt

The SME is notified of the success or failure of the MLME-PSD-Config.request operation.

3.3.3 MLME-PSD-Receive.indication

3.3.3.1 Semantics of the Service Primitive

The contents of the received discovery information element and its format identifier are passed to the MLME SAP by using the following primitive:

```
MLME-PSD-Receive.indication
(Discovery data,
FormatIdentifier)
```

3.3.3.2 When the Primitive Is Generated

The primitive is generated by the MLME upon reception of a Beacon or Probe Response. The primitive indicates that a discovery information element was contained in the Beacon or Probe Response that matches a format identifier that was previously registered by using the MLME-PSD.Config.request primitive. If the Beacon or Probe Response contains more than one such information element, multiple MLME-PSD-Receive.indication primitives will be generated.

3.3.3.3 Effect on Receipt

The primitive indicates discovery data to the SME.

3.4 Other Protocol Data

3.4.1 Abstract Data Model

The Proximity Service Discovery Protocol has no abstract data model.

3.4.2 Timers

The Proximity Service Discovery Protocol defines no timers.

3.4.3 Initialization

The discovery is activated by the start of the beacon transmission by an AP or ad hoc station and the inclusion of the information element in the Beacon frame. It terminates when the beacon transmission ceases or the discovery information element is removed from the Beacon frame.

3.4.4 Higher-Layer Triggered Events

The Proximity Service Discovery Protocol has no higher-layer triggered events.

3.4.5 Message Processing Events and Sequencing Rules

In the Proximity Service Discovery Protocol, it is the responsibility of the application to protect itself against hash collisions. For more information, see sections 3.1 and 3.2.

3.4.6 Timer Events

The Proximity Service Discovery Protocol has no timer events.

3.4.7 Other Local Events

The Proximity Service Discovery Protocol has no additional local events.

4 Protocol Examples

This section describes example operations that are used in common scenarios to illustrate the function of the Proximity Service Discovery Protocol.

Examples showing computation of format identifiers from namespace URIs:

```
String:  "http://schemas.xmlsoaps.org/ws/2004/10/discovery"
```

First (leftmost) 4 octets of HMAC–SHA256 (format id) of the previous string: 0xF8 0xCB 0x35 0x15.

```
String:  "http://schemas.microsoft.com/networking/discoveryformat/v2"
```

First (leftmost) 4 octets of HMAC–SHA256 (format id) of the previous string: 0xCF 0xF1 0x64 0x17.

Note The transmission order is: The first (leftmost) octet is transmitted first.

If the "shatest" program as specified in [RFC4634] is used to produce the format identifier from a string, the procedure is as follows:

1. Create a testfile containing the string in UTF-16 little endian format
2. Call shatest: shatest -h 2 -f testfile -k ""

The following is an example of the vendor extension IE conveyed in a Beacon or Probe Response frame for the format identifier string "test" (without quotes) and 8 octets data in the payload.

Offset (hex)	Value (hex)	Field
0000	DD	Element ID
0001	10	Length (16 octets)
0002	00	OUI (Microsoft)
0003	50	
0004	F2	
0005	06	OUI Type
0006	9C	Format Identifier hash of "test"
0007	19	
0008	EB	
0009	4A	
000A ... 0012	01 02 03 04 05 06 07 08	Data (8 octets in example)

Figure 3: Example of the vendor extension IE

5 Security

The following sections specify security considerations for implementers of the Proximity Service Discovery Protocol.

5.1 Security Considerations for Implementers

This protocol currently has no security considerations for implementers.

5.2 Index of Security Parameters

This protocol has no security parameters.

6 Appendix A: Windows Behavior

The information in this specification is applicable to the following versions of Windows:

- Windows Vista

Exceptions, if any, are noted below. Unless otherwise specified, any statement of optional behavior in this specification prescribed using the terms SHOULD or SHOULD NOT implies Windows behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that Windows does not follow the prescription.

<1> Section 1.3: The OUI Type that Microsoft has defined for this protocol is 6.

<2> Section 1.3: Microsoft does not define any data for Windows nor uses any of the capabilities of this protocol for its own purposes.

<3> Section 1.3: Windows does not use this protocol for service discovery.

<4> Section 2.1: Windows does not discard duplicate information elements.

<5> Section 3.2: Windows limits the number of information elements to 5.

7 Index