[MS-SPTWS]:

Service Platform Topology Web Service Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.

- **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 can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.

- **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.

- **Patents**. Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft Open Specifications Promise or the Microsoft Community Promise. If you would prefer a written license, or if the technologies described in this documentation 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 might 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 that are 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 as specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications documentation does 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 documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

Revision Summary

Date	Revision History	Revision Class	Comments
7/13/2009	0.1	Major	Initial Availability
8/28/2009	0.2	Editorial	Revised and edited the technical content
11/6/2009	0.3	Editorial	Revised and edited the technical content
2/19/2010	1.0	Major	Updated and revised the technical content
3/31/2010	1.01	Editorial	Revised and edited the technical content
4/30/2010	1.02	Editorial	Revised and edited the technical content
6/7/2010	1.03	Editorial	Revised and edited the technical content
6/29/2010	1.04	Editorial	Changed language and formatting in the technical content.
7/23/2010	1.05	Minor	Clarified the meaning of the technical content.
9/27/2010	1.05	None	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	1.05	None	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	1.06	Editorial	Changed language and formatting in the technical content.
3/18/2011	1.06	None	No changes to the meaning, language, or formatting of the technical content.
6/10/2011	1.06	None	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	2.0	Major	Significantly changed the technical content.
4/11/2012	2.0	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	2.0	None	No changes to the meaning, language, or formatting of the technical content.
9/12/2012	2.0	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	2.0	None	No changes to the meaning, language, or formatting of the technical content.
2/11/2013	2.0	None	No changes to the meaning, language, or formatting of the technical content.
7/30/2013	2.1	Minor	Clarified the meaning of the technical content.
11/18/2013	2.1	None	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	2.1	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	2.1	None	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
7/31/2014	2.1	None	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	2.1	None	No changes to the meaning, language, or formatting of the technical content.
2/26/2016	3.0	Major	Significantly changed the technical content.
7/15/2016	3.0	None	No changes to the meaning, language, or formatting of the technical content.
9/14/2016	3.0	None	No changes to the meaning, language, or formatting of the technical content.
12/15/2016	3.1	Minor	Clarified the meaning of the technical content.

Table of Contents

1   Introduction

The Service Platform Topology Web Service Protocol allows a protocol client to request a list of the **service applications** that are known to a protocol server, and to retrieve information, such as **endpoint URIs**, about those service applications.

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

1.1   Glossary

This document uses the following terms:

Augmented Backus-Naur Form (ABNF): A modified version of Backus-Naur Form (BNF), commonly used by Internet specifications. ABNF notation balances compactness and simplicity with reasonable representational power. ABNF differs from standard BNF in its definitions and uses of naming rules, repetition, alternatives, order-independence, and value ranges. For more information, see [RFC5234].

build number: A part of a sequential numbering system that is used to differentiate one version of a software product from another.

class module: A module that contains the definition for a new object. Each instance of a class creates a new object, and procedures that are defined in the module become properties and methods of the object.

display name: A text string that is used to identify a principal or other object in the user interface. Also referred to as title.

endpoint: A communication port that is exposed by an application server for a specific shared service and to which messages can be addressed.

globally unique identifier (GUID): A term used interchangeably with universally unique identifier (UUID) in Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the value. Specifically, the use of this term does not imply or require that the algorithms described in [RFC4122] or [C706] must be used for generating the **GUID**. See also universally unique identifier (UUID).

Hypertext Transfer Protocol (HTTP): An application-level protocol for distributed, collaborative, hypermedia information systems (text, graphic images, sound, video, and other multimedia files) on the World Wide Web.

major version: An iteration of a software component, document, or list item that is ready for a larger group to see, or has changed significantly from the previous major version. For an item on a SharePoint site, the **minor version** is always "0" (zero) for a major version.

minor version: An iteration of a software component, document, or list item that is in progress or has changed only slightly from the previous version. For an item on a SharePoint site, the minor version number is never "0" (zero) and is incremented for each new version of an item, unless a **major version** is explicitly published. When minor versioning is disabled on a SharePoint site, only major version numbers are incremented, and the minor version is always "0" (zero).

revision: A change in a document, file, or other object.

service application: A middle-tier application that runs without any user interface components and supports other applications by performing tasks such as retrieving or modifying data in a database.

SOAP action: The HTTP request header field used to indicate the intent of the SOAP request, using a **URI** value. See [SOAP1.1] section 6.1.1 for more information.

SOAP body: A container for the payload data being delivered by a SOAP message to its recipient. See [SOAP1.2-1/2007] section 5.3 for more information.

SOAP fault: A container for error and status information within a SOAP message. See [SOAP1.2-1/2007] section 5.4 for more information.

Status-Code: A 3-digit integer result code in an HTTP response message, as described in [RFC2616].

Uniform Resource Identifier (URI): A string that identifies a resource. The URI is an addressing mechanism defined in Internet Engineering Task Force (IETF) Uniform Resource Identifier (URI): Generic Syntax [RFC3986].

Uniform Resource Locator (URL): A string of characters in a standardized format that identifies a document or resource on the World Wide Web. The format is as specified in [RFC1738].

Web Services Description Language (WSDL): An XML format for describing network services as a set of endpoints that operate on messages that contain either document-oriented or procedure-oriented information. The operations and messages are described abstractly and are bound to a concrete network protocol and message format in order to define an endpoint. Related concrete endpoints are combined into abstract endpoints, which describe a network service. WSDL is extensible, which allows the description of endpoints and their messages regardless of the message formats or network protocols that are used.

WSDL message: An abstract, typed definition of the data that is communicated during a **WSDL operation** [WSDL]. Also, an element that describes the data being exchanged between web service providers and clients.

WSDL operation: A single action or function of a web service. The execution of a WSDL operation typically requires the exchange of messages between the service requestor and the service provider.

XML namespace: A collection of names that is used to identify elements, types, and attributes in XML documents identified in a URI reference [RFC3986]. A combination of XML namespace and local name allows XML documents to use elements, types, and attributes that have the same names but come from different sources. For more information, see [XMLNS-2ED].

XML namespace prefix: An abbreviated form of an **XML namespace**, as described in [XML].

XML schema: A description of a type of XML document that is typically expressed in terms of constraints on the structure and content of documents of that type, in addition to the basic syntax constraints that are imposed by XML itself. An XML schema provides a view of a document type at a relatively high level of abstraction.

XML schema definition (XSD): The World Wide Web Consortium (W3C) standard language that is used in defining XML schemas. Schemas are useful for enforcing structure and constraining the types of data that can be used validly within other XML documents. XML schema definition refers to the fully specified and currently recommended standard for use in authoring **XML schemas**.

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

1.2   References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents

in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

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.

[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

[RFC2616] Fielding, R., Gettys, J., Mogul, J., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, http://www.rfc-editor.org/rfc/rfc2616.txt

[SOAP1.1] Box, D., Ehnebuske, D., Kakivaya, G., et al., "Simple Object Access Protocol (SOAP) 1.1", W3C Note, May 2000, http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

[WSA1.0] Gudgin, M., Hadley, M., Rogers, T., et al., Eds., "Web Services Addressing 1.0 - WSDL Binding", W3C Candidate Recommendation, May 2006, http://www.w3.org/TR/2006/CR-ws-addr-wsdl-20060529/

[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., 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., Ed. and Malhotra, A., Ed., "XML Schema Part 2: Datatypes", W3C Recommendation, May 2001, http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/

1.2.2 Informative References

None.

1.3 Protocol Overview (Synopsis)

This protocol allows a protocol client to obtain a list of information about **service applications** that are known to a protocol server. It also allows the protocol client to retrieve a list of **endpoint Uniform Resource Identifiers (URI)** for a service application and to communicate with that service application by using a protocol supported by that service application.

This protocol supports two major types of operations:

▪ Enumerating service applications **-** A protocol client can request a list of service applications that are known to a protocol server. The protocol client can display this list to a user and enable the user to select and perform additional operations with a service application that is in the list.

▪ Retrieving endpoints for a service application **-** A protocol client can request a list of endpoint URIs for a specified service application. Protocol clients typically use this type of operation to identify the endpoints of the protocols that they are using.

1.4 Relationship to Other Protocols

This protocol uses the SOAP message protocol for formatting request and response messages, as described in [SOAP1.1], [SOAP1.2/1] and [SOAP1.2/2]. It transmits those messages by using HTTPS, as described in [RFC2818].

The following diagram shows the underlying messaging and transport stack used by the protocol.
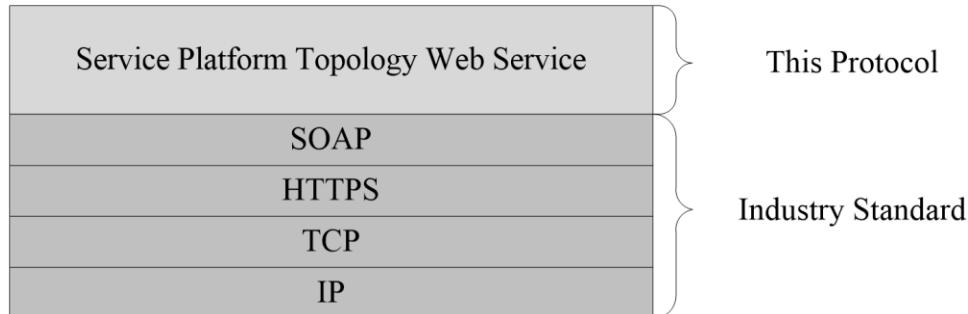
Figure 1: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

This protocol operates against a protocol server that exposes one or more **endpoint URIs** that are known by protocol clients. The protocol server endpoint URI is known by the protocol client.

This protocol requires that the protocol client has the requisite permission to call methods that are implemented on the protocol server.

1.6 Applicability Statement

This protocol is intended for use by protocol clients and protocol servers that are connected by high-bandwidth, low-latency network connections.

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- **Supported Transports**: This protocol can be implemented by using transports that support sending Simple Object Access Protocol (SOAP) messages, as described in section 2.1.

- **Protocol Versions**: This protocol is not versioned.

Capability Negotiation: This protocol does not support version negotiation.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

In the following sections, the schema definition might differ from the processing rules imposed by the protocol. The **WSDL** in this specification matches the WSDL that shipped with the product and provides a base description of the schema. The text that introduces the WSDL might specify differences that reflect actual Microsoft product behavior. For example, the schema definition might allow for an element to be **empty**, **null**, or **not present** but the behavior of the protocol as specified restricts the same elements to being **non-empty**, **not null**, and **present**.

2.1 Transport

Protocol servers MUST support SOAP over HTTPS.

All protocol messages MUST be transported by using HTTPS bindings at the transport level and a server certificate MUST be deployed.

Protocol messages MUST be formatted as specified in either [SOAP1.1] section 4 or [SOAP1.2/1] section 5. Protocol server faults MUST be returned by using either **HTTP** status codes, as specified in [RFC2616] section 10, or **SOAP faults**, as specified in [SOAP1.1] section 4.4 or [SOAP1.2/1] section 5.4.

This protocol MAY transmit an additional SOAP header, the **ServiceContext** header, as specified in [MS-SPSTWS].

This protocol does not define any means for activating a protocol server or protocol client. The protocol server MUST be configured and begin listening in an implementation-specific way.

2.2 Common Message Syntax

This section contains common definitions that are used by this protocol. The syntax of the definitions uses **XML schema**, as specified in [XMLSCHEMA1] and [XMLSCHEMA2], and **WSDL**, as specified in [WSDL].

2.2.1 Namespaces

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

Prefix	Namespace URI	Reference
q1	http://schemas.datacontract.org/2004/07/Microsoft.SharePoint	
q2	http://schemas.datacontract.org/2004/07/System	
ser	http://schemas.microsoft.com/2003/10/Serialization/	
soap	http://schemas.xmlsoap.org/wsdl/soap/	[SOAP1.1]
tns	http://tempuri.org/	
tns1	http://schemas.datacontract.org/2004/07/Microsoft.SharePoint.Administration	
tns2	http://schemas.microsoft.com/2003/10/Serialization/Arrays	
tns3	http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel	

Prefix	Namespace URI	Reference
tns4	http://tempuri.org/Imports	
wsaw	http://www.w3.org/2006/05/addressing/wsdl	[WSA1.0]
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1] [XMLSCHEMA2]

2.2.2 Messages

This specification does not define any common **WSDL message** definitions.

2.2.3 Elements

This specification does not define any common **XML schema** element definitions.

2.2.4 Complex Types

The following table summarizes the set of common **XML schema** complex type definitions defined by this specification. XML schema complex type definitions that are specific to a particular operation are described with the operation.

Complex type	Description
SPTopologyWebServiceApplicationFault	The **SPTopologyWebServiceApplicationFault** element specifies a **SOAP fault**.

2.2.4.1 SPTopologyWebServiceApplicationFault

Namespace: http://schemas.datacontract.org/2004/07/Microsoft.SharePoint

The **SPTopologyWebServiceApplicationFault** element specifies a **SOAP fault**.

```
<xs:complexType name="SPTopologyWebServiceApplicationFault"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" name="FaultReason" nillable="true" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

FaultReason: A string specifying the reason for the fault. This value MUST be ignored by the protocol client.

2.2.5 Simple Types

This specification does not define any common **XML schema** simple type definitions.

2.2.6 Attributes

This specification does not define any common **XML schema** attribute definitions.

2.2.7 Groups

This specification does not define any common **XML schema** group definitions.

2.2.8 Attribute Groups

This specification does not define any common **XML schema** attribute group definitions.

3   Protocol Details

In the following sections, the schema definition might differ from the processing rules imposed by the protocol. The **WSDL** in this specification matches the WSDL that shipped with the product and provides a base description of the schema. The text that introduces the WSDL might specify differences that reflect actual Microsoft product behavior. For example, the schema definition might allow for an element to be **empty**, **null**, or **not present** but the behavior of the protocol as specified restricts the same elements to being **non-empty**, **not null**, and **present**.

The client side of this protocol is simply a pass-through. That is, no additional timers or other state is required on the client side of this protocol. Calls made by the higher-layer protocol or application are passed directly to the transport, and the results returned by the transport are passed directly back to the higher-layer protocol or application.

This protocol allows protocol servers to perform implementation-specific authorization checks and to notify protocol clients of authorization faults by using **HTTP Status-Codes** or **SOAP faults**. Except where specified otherwise, protocol clients SHOULD interpret HTTP Status-Codes as specified in [RFC2616] section 10. Protocol servers can notify protocol clients of application-level faults by using SOAP faults. Except where specified otherwise, these SOAP faults are not significant for interoperability and protocol clients can interpret them in an implementation-specific manner.

3.1   ITopologyWebServiceApplication Server Details

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 document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

The protocol server maintains a list of **service applications** and the following properties for each of those service applications:

- **GUID -** A unique identifier that represents the service application.

- **ApplicationClassId -** A unique identifier that represents the service application **class module**.

- **ApplicationVersion -** The version number of the service application class module.

- **Comments -** A string that contains descriptive information about the service application.

- **DisplayName -** A string that specifies the **display name** of the service application.

- **TermsOfServiceUri -** A **Uniform Resource Locator (URL)** for a document that contains additional, descriptive information about the service application.

- **Uri -** A **URI** that specifies a single address that logically represents the service application.

The format of these properties is specified in section 3.1.4.1.3.3.

The protocol server also maintains information about its **endpoint** URIs in this list.

endpoint URIs **-** A list of all the endpoint URIs for the service application. The protocol server maintains a list of endpoint URIs for the default transport that the service application supports. In response to messages from a protocol client, the protocol server sends a list of service applications and their properties to the protocol client. The addition, deletion, and modification of the list of service applications and their properties are not specified by this protocol and are implementation-specific.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Message Processing Events and Sequencing Rules

The following table summarizes the list of operations as defined by this specification.

Operation	Description
EnumerateSharedServiceApplications	The **EnumerateSharedServiceApplications** operation returns a collection of information about all of the **service applications** on the protocol server.
GetEndPoints	The **GetEndPoints** operation returns a collection of all the **endpoint URIs** for a service application that is specified by a **GUID** in an input message.

3.1.4.1 EnumerateSharedServiceApplications

The **EnumerateSharedServiceApplications** operation returns a collection of information about all of the **service applications** on the protocol server.

The following is the **WSDL** port type specification of the **EnumerateSharedServiceApplications WSDL operation**.

```
<wsdl:operation name="EnumerateSharedServiceApplications"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input
wsaw:Action="http://tempuri.org/ITopologyWebServiceApplication/EnumerateSharedServiceApplicat
ions"
message="tns:ITopologyWebServiceApplication EnumerateSharedServiceApplications InputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output
wsaw:Action="http://tempuri.org/ITopologyWebServiceApplication/EnumerateSharedServiceApplicat
ionsResponse"
message="tns:ITopologyWebServiceApplication_EnumerateSharedServiceApplications_OutputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:fault
wsaw:Action="http://tempuri.org/ITopologyWebServiceApplication/EnumerateSharedServiceApplicat
ionsSPTopologyWebServiceApplicationFaultFault"
name="SPTopologyWebServiceApplicationFaultFault"
message="tns:ITopologyWebServiceApplication_EnumerateSharedServiceApplications_SPTopologyWebS
erviceApplicationFaultFault FaultMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>
```

The protocol client sends an
ITopologyWebServiceApplication_EnumerateSharedServiceApplications_InputMessage
request **WSDL message** and the protocol server MUST respond with an
ITopologyWebServiceApplication_EnumerateSharedServiceApplications_OutputMessage
response WSDL message, as follows:

- If one or more service applications exist on the protocol server, the protocol server MUST return a collection of information about all of those service applications. The information is contained in an **EnumerateSharedServiceApplicationsResult** element, as specified in section 3.1.4.1.2.2.

- If no service applications exist on the protocol server, the protocol server MUST return an empty collection.

3.1.4.1.1 Messages

The following table summarizes the set of **WSDL message** definitions that are specific to this operation.

Message	Description
ITopologyWebServiceApplication_EnumerateSharedServiceApplications_InputMessage	The request WSDL message for the **EnumerateSharedServiceApplications WSDL operation**.
ITopologyWebServiceApplication_EnumerateSharedServiceApplications_OutputMessage	The response WSDL message for the **EnumerateSharedServiceApplications** WSDL operation.

3.1.4.1.1.1   ITopologyWebServiceApplication_EnumerateSharedServiceApplications_ InputMessage

The request **WSDL message** for the **EnumerateSharedServiceApplications WSDL operation**.

The **SOAP action** value is:

```
http://tempuri.org/ITopologyWebServiceApplication/EnumerateSharedServiceApplications
```

The **SOAP body** contains the **EnumerateSharedServiceApplications** element.

3.1.4.1.1.2   ITopologyWebServiceApplication_EnumerateSharedServiceApplications_ OutputMessage

The response **WSDL message** for the **EnumerateSharedServiceApplications WSDL operation**.

The **SOAP body** contains the **EnumerateSharedServiceApplicationsResponse** element.

3.1.4.1.2 Elements

The following table summarizes the **XML schema** element definitions that are specific to this operation.

Element	Description
EnumerateSharedServiceApplications	The input data for the **EnumerateSharedServiceApplications WSDL operation**.
EnumerateSharedServiceApplicationsResponse	The result data for the **EnumerateSharedServiceApplications** WSDL operation.

3.1.4.1.2.1   EnumerateSharedServiceApplications

The **EnumerateSharedServiceApplications** element specifies the input data for the
EnumerateSharedServiceApplications WSDL operation.

```
<xs:element name="EnumerateSharedServiceApplications"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
```

3.1.4.1.2.2   EnumerateSharedServiceApplicationsResponse

The **EnumerateSharedServiceApplicationsResponse** element specifies the result data for the
EnumerateSharedServiceApplications WSDL operation.

```
<xs:element name="EnumerateSharedServiceApplicationsResponse"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element
xmlns:tns3="http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel"
minOccurs="0" name="EnumerateSharedServiceApplicationsResult" nillable="true"
type="tns3:ReadOnlyCollectionOfSPSharedServiceApplicationInfo1RC_ScQDg"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

EnumerateSharedServiceApplicationsResult: A collection of information about all of the **service
applications** that are on the protocol server. It MUST conform to the **XML schema definition (XSD)**
of the **ReadOnlyCollectionOfSPSharedServiceApplicationInfo1RC_ScQDg** complex type, as
specified in section 3.1.4.1.3.1. This element MUST be present and MUST NOT be empty.

3.1.4.1.3 Complex Types

The following table summarizes the **XML schema** complex type definitions that are specific to this
operation.

Complex type	Description
ArrayOfSPSharedServiceApplicationInfo	The **ArrayOfSPSharedServiceApplicationInfo** complex type is a list of objects, each of which contains information about a specific **service application**.
ReadOnlyCollectionOfSPSharedServiceApplication Info1RC_ScQDg	The **ReadOnlyCollectionOfSPSharedServiceApplication Info1RC_ScQDg** complex type contains the data that results from an **EnumerateSharedServiceApplications WSDL operation**.
SPSharedServiceApplicationInfo	The **SPSharedServiceApplicationInfo** complex type contains information about a service application.
Version	The **Version** complex type is a structure that contains version information for a service application **class module**.

3.1.4.1.3.1 ReadOnlyCollectionOfSPSharedServiceApplicationInfo1RC_ScQDg

Namespace: http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel

The **ReadOnlyCollectionOfSPSharedServiceApplicationInfo1RC_ScQDg** complex type contains the data that results from an **EnumerateSharedServiceApplications WSDL operation**. It is defined by the namespace at http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel.

```
<xs:complexType name="ReadOnlyCollectionOfSPSharedServiceApplicationInfo1RC_ScQDg"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element
xmlns:tns1="http://schemas.datacontract.org/2004/07/Microsoft.SharePoint.Administration"
name="list" nillable="true" type="tns1:ArrayOfSPSharedServiceApplicationInfo"/>
  </xs:sequence>
</xs:complexType>
```

list: An array of **SPSharedServiceApplicationInfo**. It MUST conform to the **XSD** of the **ArrayOfSPSharedServiceApplicationInfo** complex type, as specified in section 3.1.4.1.3.2.

3.1.4.1.3.2 ArrayOfSPSharedServiceApplicationInfo

Namespace: http://schemas.datacontract.org/2004/07/Microsoft.SharePoint.Administration

The **ArrayOfSPSharedServiceApplicationInfo** complex type is a list of **SPSharedServiceApplicationInfo** objects, each of which contains information about a specific **service application**. It is defined by the namespace at http://schemas.datacontract.org/2004/07/Microsoft.SharePoint.Administration. If there are no service applications on the server, the returned list MUST be empty.

```
<xs:complexType name="ArrayOfSPSharedServiceApplicationInfo"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="SPSharedServiceApplicationInfo"
nillable="true" type="tns1:SPSharedServiceApplicationInfo"/>
  </xs:sequence>
</xs:complexType>
```

SPSharedServiceApplicationInfo: A structure that contains information about the service application. It MUST conform to the **XSD** of the **SPSharedServiceApplicationInfo** complex type, as specified in section 3.1.4.1.3.3.

3.1.4.1.3.3 SPSharedServiceApplicationInfo

Namespace: http://schemas.datacontract.org/2004/07/Microsoft.SharePoint.Administration

The **SPSharedServiceApplicationInfo** complex type contains information about a **service application**. It is defined by the namespace at http://schemas.datacontract.org/2004/07/Microsoft.SharePoint.Administration.

```
<xs:complexType name="SPSharedServiceApplicationInfo"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" name="ApplicationClassId" type="ser:guid"/>
    <xs:element xmlns:q2="http://schemas.datacontract.org/2004/07/System" minOccurs="0"
name="ApplicationVersion" nillable="true" type="q2:Version"/>
```

```
       <xs:element minOccurs="0" name="Comments" nillable="true" type="xs:string"/>
       <xs:element minOccurs="0" name="DisplayName" nillable="true" type="xs:string"/>
       <xs:element minOccurs="0" name="TermsOfServiceUri" nillable="true" type="xs:anyURI"/>
       <xs:element minOccurs="0" name="Uri" nillable="true" type="xs:anyURI"/>
    </xs:sequence>
  </xs:complexType>
```

ApplicationClassId: A **GUID** that identifies the service application **class module**. It MUST conform to the **XSD** of the **guid** simple type. This element MUST be present.

ApplicationVersion: The version of the service application class module. It MUST conform to the **XML schema** of the **Version** complex type, as specified in section [3.1.4.1.3.4](#). This element MUST be present.

Comments: A string that contains a description of the service application. This element MUST be present.

DisplayName: A string that specifies the **display name** of the service application. This element MUST be present.

TermsOfServiceUri: A **URL** that specifies a document containing additional information about the service application. This element MUST be present.

Uri: A **URI** that specifies the logical address of the service application. It MUST be either a hardware load-balancer address of the service or conform to the following **Augmented Backus-Naur Form (ABNF)**. This element MUST be present.

```
logicaladdress = "urn:schemas-microsoft-
com:sharepoint:service:"serviceid"#authority=urn:uuid:"topologyserviceid"&authority="endpoint
uri
topologyserviceid = guid
serviceid = guid
guid = 32*hexdig
hexdig = %x30-39/"A"/"B"/"C"/"D"/"E"/"F"/"a"/"b"/"c"/"d"/"e"/"f"
```

endpointuri: The **endpoint** URI of the service that is providing the implementation of this protocol.

topologyserviceid: A GUID that specifies the service that is providing the implementation of this protocol.

serviceid: A GUID that identifies the service application about which information is being returned.

3.1.4.1.3.4   Version

Namespace: http://schemas.datacontract.org/2004/07/System

The **Version** complex type is a structure that contains version information for a **service application class module**. It is defined by the namespace at http://schemas.datacontract.org/2004/07/System.

```
  <xs:complexType name="Version" xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:sequence>
      <xs:element name=" Build" type="xs:int"/>
      <xs:element name="_Major" type="xs:int"/>
      <xs:element name=" Minor" type="xs:int"/>
      <xs:element name="_Revision" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
```

_Build: An integer that specifies the **build number** of the class module. This value MUST be greater than or equal to 0.

_Major: An integer that specifies the **major version** number of the class module. This value MUST be greater than or equal to 0.

_Minor: An integer that specifies the **minor version** number of the class module. This value MUST be greater than or equal to 0.

_Revision: An integer that specifies the **revision** number of the class module. This value MUST be greater than or equal to 0.

3.1.4.1.4 Simple Types

None.

3.1.4.1.5 Attributes

None.

3.1.4.1.6 Groups

None.

3.1.4.1.7 Attribute Groups

None.

3.1.4.2 GetEndPoints

The **GetEndPoints** operation returns a collection of all the **endpoint URIs** for a **service application** that is specified by a **GUID** in an input message.

The following is the **WSDL** port type specification of the **GetEndPoints WSDL operation**.

```
<wsdl:operation name="GetEndPoints" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input wsaw:Action="http://tempuri.org/ITopologyWebServiceApplication/GetEndPoints"
message="tns:ITopologyWebServiceApplication_GetEndPoints_InputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output
wsaw:Action="http://tempuri.org/ITopologyWebServiceApplication/GetEndPointsResponse"
message="tns:ITopologyWebServiceApplication_GetEndPoints_OutputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:fault
wsaw:Action="http://tempuri.org/ITopologyWebServiceApplication/GetEndPointsSPTopologyWebServi
ceApplicationFaultFault" name="SPTopologyWebServiceApplicationFaultFault"
message="tns:ITopologyWebServiceApplication_GetEndPoints_SPTopologyWebServiceApplicationFault
Fault_FaultMessage" xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>
```

The protocol client sends an **ITopologyWebServiceApplication_GetEndPoints_InputMessage** request **WSDL message** and the protocol server MUST respond with an **ITopologyWebServiceApplication_GetEndPoints_OutputMessage** response WSDL message, as follows:

▪ If a service application with the specified GUID exists, the protocol server MUST return a collection of all the endpoint URIs for the service application.

- If a service application with the specified GUID does not exist, the protocol server MUST return a **SPTopologyWebServiceApplicationFault SOAP fault** with the **FaultReason** indicating that the service application was not found on the server.

3.1.4.2.1 Messages

The following table summarizes the set of **WSDL message** definitions that are specific to this operation.

Message	Description
ITopologyWebServiceApplication_GetEndPoints_InputMessage	The request WSDL message for the **GetEndPoints WSDL operation**.
ITopologyWebServiceApplication_GetEndPoints_OutputMessage	The response WSDL message for the **GetEndPoints** WSDL operation.

3.1.4.2.1.1   ITopologyWebServiceApplication_GetEndPoints_InputMessage

The request **WSDL message** for the **GetEndPoints WSDL operation**.

The **SOAP action** value is:

```
http://tempuri.org/ITopologyWebServiceApplication/GetEndPoints
```

The **SOAP body** contains the **GetEndPoints** element.

3.1.4.2.1.2   ITopologyWebServiceApplication_GetEndPoints_OutputMessage

The response **WSDL message** for the **GetEndPoints WSDL operation**.

The **SOAP body** contains the **GetEndPointsResponse** element.

3.1.4.2.2 Elements

The following table summarizes the **XML schema** element definitions that are specific to this operation.

Element	Description
GetEndPoints	The input data for the **GetEndPoints WSDL operation**.
GetEndPointsResponse	The result data for the **GetEndPoints** WSDL operation.

3.1.4.2.2.1   GetEndPoints

The **GetEndPoints** element specifies the input data for the **GetEndPoints WSDL operation**.

```
<xs:element name="GetEndPoints" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element xmlns:ser="http://schemas.microsoft.com/2003/10/Serialization/"
minOccurs="0" name="serviceId" type="ser:guid"/>
    </xs:sequence>
```

```
        </xs:complexType>
    </xs:element>
```

serviceId: A **GUID** that identifies the **service application**. This value MUST be present and MUST NOT be empty.

3.1.4.2.2   GetEndPointsResponse

The **GetEndPointsResponse** element specifies the result data for the **GetEndPoints WSDL operation**.

```
    <xs:element name="GetEndPointsResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:complexType>
        <xs:sequence>
          <xs:element
    xmlns:tns3="http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel"
    minOccurs="0" name="GetEndPointsResult" nillable="true"
    type="tns3:ReadOnlyCollectionOfanyURI"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
```

GetEndPointsResult: A collection of **endpoint URIs** for the specified **service application**. It MUST conform to the **XSD** of the **ReadOnlyCollectionOfanyURI** complex type, as specified in section 3.1.4.2.3.1. This element MUST be present and MUST NOT be empty.

3.1.4.2.3 Complex Types

The following table summarizes the **XML schema** complex type definitions that are specific to this operation.

Complex type	Description
ArrayOfanyURI	The **ArrayOfanyURI** complex type is an array of **endpoint URIs** for a **service application**. It is defined by the namespace at http://schemas.microsoft.com/2003/10/Serialization/Arrays.
ReadOnlyCollectionOfanyURI	The **ReadOnlyCollectionOfanyURI** complex type is a collection of endpoint URIs for a service application.

3.1.4.2.3.1   ReadOnlyCollectionOfanyURI

Namespace: http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel

The **ReadOnlyCollectionOfanyURI** complex type is a collection of **endpoint URIs** for a **service application**. It is defined by the namespace at http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel.

```
    <xs:complexType name="ReadOnlyCollectionOfanyURI"
    xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:sequence>
        <xs:element xmlns:tns2="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
    name="list" nillable="true" type="tns2:ArrayOfanyURI"/>
      </xs:sequence>
    </xs:complexType>
```

list: An array of endpoint URIs for the service application. It MUST conform to the **XSD** of the **ArrayOfanyURI** complex type, as specified in section 3.1.4.2.3.2.

3.1.4.2.3.2   ArrayOfanyURI

Namespace: http://schemas.microsoft.com/2003/10/Serialization/Arrays

The **ArrayOfanyURI** complex type is an array of **endpoint URIs** for a **service application**. It is defined by the namespace at http://schemas.microsoft.com/2003/10/Serialization/Arrays. If a service application does not have any endpoint URIs, the **ArrayOfanyURI** element MUST be empty.

```
<xs:complexType name="ArrayOfanyURI" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="anyURI" nillable="true"
type="xs:anyURI"/>
  </xs:sequence>
</xs:complexType>
```

anyURI: A specific endpoint URI for the service application.

3.1.4.2.4 Simple Types

None.

3.1.4.2.5 Attributes

None.

3.1.4.2.6 Groups

None.

3.1.4.2.7 Attribute Groups

None.

3.1.5   Timer Events

None.

3.1.6   Other Local Events

None.

4 Protocol Examples

4.1 EnumerateSharedServiceApplications Operation

In this example, a protocol client requests a list of **service application**s that are on a protocol server and the protocol server responds to that request.

The protocol client constructs the following **WSDL** request message:

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  <s:Body>
    <EnumerateSharedServiceApplications xmlns="http://tempuri.org/">
    </EnumerateSharedServiceApplications>
  </s:Body>
</s:Envelope>
```

There are three service applications on the protocol server. Therefore, the protocol server responds with the following **WSDL message**:

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  <s:Body>
    <EnumerateSharedServiceApplicationsResponse xmlns="http://tempuri.org/">
      <EnumerateSharedServiceApplicationsResult
xmlns:b="http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <b:list
xmlns:c="http://schemas.datacontract.org/2004/07/Microsoft.SharePoint.Administration">
          <c:SPSharedServiceApplicationInfo>
            <c:ApplicationClassId>e8479529-b61f-410a-a631-11e577975716</c:ApplicationClassId>
            <c:ApplicationVersion xmlns:d="http://schemas.datacontract.org/2004/07/System">
              <d:_Build>0</d:_Build>
              <d:_Major>1</d:_Major>
              <d:_Minor>0</d:_Minor>
              <d:_Revision>0</d:_Revision>
            </c:ApplicationVersion>
            <c:Comments>Service1 provides functionality1</c:Comments>
            <c:DisplayName>Service1App</c:DisplayName>
            <c:TermsOfServiceUri
i:nil="true">http://ServerA/Service1Help.html</c:TermsOfServiceUri>
            <c:Uri>urn:schemas-microsoft-
com:sharepoint:service:deaee7d4345744b6bc7f8370ff1c8801#authority=urn:uuid:e3f6969562ad47ea91
22638e9ab488b7&amp;authority=https%3A%2F%2FServerA%3A32844%2FTopology%2Ftopology.svc</c:Uri>
          </c:SPSharedServiceApplicationInfo>
          <c:SPSharedServiceApplicationInfo>
            <c:ApplicationClassId>8801a560-b15b-41e5-a661-e1ac85bdad9e</c:ApplicationClassId>
            <c:ApplicationVersion xmlns:d="http://schemas.datacontract.org/2004/07/System">
              <d:_Build>0</d:_Build>
              <d:_Major>1</d:_Major>
              <d:_Minor>0</d:_Minor>
              <d:_Revision>0</d:_Revision>
            </c:ApplicationVersion>
            <c:Comments>Service2 provides functionality2</c:Comments>
            <c:DisplayName>Service2App</c:DisplayName>
            <c:TermsOfServiceUri i:nil="true">http://ServerA/Service2Help.html
</c:TermsOfServiceUri>
            <c:Uri>urn:schemas-microsoft-com:sharepoint:service:
7b59f9413c534e6883d76eee5ee0125c#authority=urn:uuid:e3f6969562ad47ea9122638e9ab488b7&amp;auth
ority=https%3A%2F%2FServerA%3A32844%2FTopology%2Ftopology.svc</c:Uri>
          </c:SPSharedServiceApplicationInfo>
          <c:SPSharedServiceApplicationInfo>
            <c:ApplicationClassId>9901a560-b15b-41e5-a661-e1ac85bdad9e</c:ApplicationClassId>
            <c:ApplicationVersion xmlns:d="http://schemas.datacontract.org/2004/07/System">
              <d:_Build>0</d:_Build>
              <d:_Major>1</d:_Major>
```

```
            <d:_Minor>0</d:_Minor>
            <d:_Revision>0</d:_Revision>
        </c:ApplicationVersion>
        <c:Comments>Service3 provides functionality3</c:Comments>
        <c:DisplayName>Service3App</c:DisplayName>
        <c:TermsOfServiceUri i:nil="true"> http://ServerA/Service3Help.html
</c:TermsOfServiceUri>
        <c:Uri>urn:schemas-microsoft-
com:sharepoint:service:cc5de64c76a54b129fa7e35c5124be49#authority=urn:uuid:e3f6969562ad47ea91
22638e9ab488b7&amp;authority=https%3A%2F%2FServerA%3A32844%2FTopology%2Ftopology.svc</c:Uri>
        </c:SPSharedServiceApplicationInfo>
      </b:list>
    </EnumerateSharedServiceApplicationsResult>
  </EnumerateSharedServiceApplicationsResponse>
  </s:Body>
</s:Envelope>
```

The protocol client then uses the data in the **ApplicationClassId** and **ApplicationVersion** elements in the response message to identify the service application with which it can communicate. The protocol client extracts the **serviceId** element from the **Uri** element for a service application and uses that information to retrieve a list of **endpoint URIs** for the service application as demonstrated in section 4.2.

If there were zero service application on the protocol server, the protocol server would have responded with the following message:

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  <s:Body>
    <EnumerateSharedServiceApplicationsResponse xmlns="http://tempuri.org/">
      <EnumerateSharedServiceApplicationsResult
xmlns:b="http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <b:list
xmlns:c="http://schemas.datacontract.org/2004/07/Microsoft.SharePoint.Administration"></b:lis
t>
      </EnumerateSharedServiceApplicationsResult>
    </EnumerateSharedServiceApplicationsResponse>
  </s:Body>
</s:Envelope>
```

4.2   GetEndPoints Operation

In this example, a protocol client retrieves all the **endpoint URIs** for a **service application** on a protocol server. It does so by extracting the value of the **topologyserviceid** parameter, which is "e3f69695-62ad-47ea-9122-638e9ab488b7", from the **Uri** element of a service application that was returned in response to an **EnumerateSharedServiceApplications** message.

To retrieve all the endpoint URIs for the service application with serviceid e3f69695-62ad-47ea-9122-638e9ab488b7 that implements this protocol, the protocol client constructs the following **WSDL message**:

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  <s:Body>
    <GetEndPoints xmlns="http://tempuri.org/">
      <serviceId>e3f69695-62ad-47ea-9122-638e9ab488b7</serviceId>
    </GetEndPoints>
  </s:Body>
</s:Envelope>
```

A service application with the specified **GUID** exists. Therefore, the protocol server responds with the following WSDL message:

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  <s:Body>
    <GetEndPointsResponse xmlns="http://tempuri.org/">
      <GetEndPointsResult
xmlns:b="http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <b:list xmlns:c="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
<c:anyURI>https://ServerA:32844/Topology/Topology.svc</c:anyURI>
        </b:list>
      </GetEndPointsResult>
    </GetEndPointsResponse>
  </s:Body>
</s:Envelope>
```

As described in section 4.1, the **EnumerateSharedServiceApplicationsResponse** message contains **ApplicationClassId** and **ApplicationVersion** elements that provide information about the service application. The protocol client uses the values in the **ApplicationClassId** and **ApplicationVersion** elements to determine whether it can communicate with the service application. The protocol client also extracts the **serviceId** value from the **Uri** element in the response. The protocol client uses the **serviceId** value to construct the following WSDL message and retrieve a read-only collection of all the endpoint URIs for the service application:

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  <s:Body>
    <GetEndPoints xmlns="http://tempuri.org/">
      <serviceId>cc5de64c-76a5-4b12-9fa7-e35c5124be49</serviceId>
    </GetEndPoints>
  </s:Body>
</s:Envelope>
```

The service application with the specified GUID exists. Therefore, the protocol server responds with the following WSDL message:

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  <s:Body>
    <GetEndPointsResponse xmlns="http://tempuri.org/">
      <GetEndPointsResult
xmlns:b="http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <b:list xmlns:c="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
<c:anyURI>http://ServerA:32844/cc5de64c76a54b129fa7e35c5124be49/Service1.svc</c:anyURI>
<c:anyURI>https://ServerA:32844/cc5de64c76a54b129fa7e35c5124be49/Service1.svc</c:anyURI>
        </b:list>
      </GetEndPointsResult>
    </GetEndPointsResponse>
  </s:Body>
</s:Envelope>
```

If the service application with the specified GUID did not exist, the protocol server would send the following response instead:

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  <s:Body>
    <s:Fault>
      <s:Code>
        <s:Value>s:Sender</s:Value>
      </s:Code>
```

```
        <s:Reason>
          <s:Text xml:lang="en-US">The creator of this fault did not specify a Reason.</s:Text>
        </s:Reason>
        <s:Detail>
          <SPTopologyWebServiceApplicationFault
xmlns="http://schemas.datacontract.org/2004/07/Microsoft.SharePoint"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
            <FaultReason>The requested application could not be found.</FaultReason>
          </SPTopologyWebServiceApplicationFault>
        </s:Detail>
      </s:Fault>
    </s:Body>
</s:Envelope>
```

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

6 Appendix A: Full WSDL

For ease of implementation, the full WSDL is provided in this appendix.

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://tempuri.org/"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl" targetNamespace="http://tempuri.org/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <xs:schema xmlns:tns4="http://tempuri.org/Imports"
targetNamespace="http://tempuri.org/Imports">
      <xs:import namespace="http://schemas.datacontract.org/2004/07/Microsoft.SharePoint"/>
      <xs:import
namespace="http://schemas.datacontract.org/2004/07/Microsoft.SharePoint.Administration"/>
      <xs:import namespace="http://schemas.datacontract.org/2004/07/System"/>
      <xs:import
namespace="http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel"/>
      <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/"/>
      <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
      <xs:import namespace="http://tempuri.org/"/>
    </xs:schema>
  </wsdl:types>
  <wsdl:portType name="ITopologyWebServiceApplication">
    <wsdl:operation name="GetEndPoints">
      <wsdl:input
wsaw:Action="http://tempuri.org/ITopologyWebServiceApplication/GetEndPoints"
message="tns:ITopologyWebServiceApplication_GetEndPoints_InputMessage"/>
      <wsdl:output
wsaw:Action="http://tempuri.org/ITopologyWebServiceApplication/GetEndPointsResponse"
message="tns:ITopologyWebServiceApplication_GetEndPoints_OutputMessage"/>
      <wsdl:fault
wsaw:Action="http://tempuri.org/ITopologyWebServiceApplication/GetEndPointsSPTopologyWebServi
ceApplicationFaultFault" name="SPTopologyWebServiceApplicationFaultFault"
message="tns:ITopologyWebServiceApplication_GetEndPoints_SPTopologyWebServiceApplicationFault
Fault_FaultMessage"/>
    </wsdl:operation>
    <wsdl:operation name="EnumerateSharedServiceApplications">
      <wsdl:input
wsaw:Action="http://tempuri.org/ITopologyWebServiceApplication/EnumerateSharedServiceApplicat
ions"
message="tns:ITopologyWebServiceApplication_EnumerateSharedServiceApplications_InputMessage"/
>
      <wsdl:output
wsaw:Action="http://tempuri.org/ITopologyWebServiceApplication/EnumerateSharedServiceApplicat
ionsResponse"
message="tns:ITopologyWebServiceApplication_EnumerateSharedServiceApplications_OutputMessage"
/>
      <wsdl:fault
wsaw:Action="http://tempuri.org/ITopologyWebServiceApplication/EnumerateSharedServiceApplicat
ionsSPTopologyWebServiceApplicationFaultFault"
name="SPTopologyWebServiceApplicationFaultFault"
message="tns:ITopologyWebServiceApplication_EnumerateSharedServiceApplications_SPTopologyWebS
erviceApplicationFaultFault_FaultMessage"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="DefaultBinding_ITopologyWebServiceApplication"
type="tns:ITopologyWebServiceApplication">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="GetEndPoints">
      <soap:operation
soapAction="http://tempuri.org/ITopologyWebServiceApplication/GetEndPoints"
style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
```

```
                <soap:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="SPTopologyWebServiceApplicationFaultFault">
                <soap:fault use="literal" name="SPTopologyWebServiceApplicationFaultFault"
namespace=""/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="EnumerateSharedServiceApplications">
            <soap:operation
soapAction="http://tempuri.org/ITopologyWebServiceApplication/EnumerateSharedServiceApplicati
ons" style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="SPTopologyWebServiceApplicationFaultFault">
                <soap:fault use="literal" name="SPTopologyWebServiceApplicationFaultFault"
namespace=""/>
            </wsdl:fault>
        </wsdl:operation>
    </wsdl:binding>
    <wsdl:message
name="ITopologyWebServiceApplication_EnumerateSharedServiceApplications_InputMessage">
        <wsdl:part name="parameters" element="tns:EnumerateSharedServiceApplications"/>
    </wsdl:message>
    <wsdl:message
name="ITopologyWebServiceApplication_EnumerateSharedServiceApplications_OutputMessage">
        <wsdl:part name="parameters" element="tns:EnumerateSharedServiceApplicationsResponse"/>
    </wsdl:message>
    <wsdl:message
name="ITopologyWebServiceApplication_EnumerateSharedServiceApplications_SPTopologyWebServiceA
pplicationFaultFault_FaultMessage">
        <wsdl:part xmlns:q1="http://schemas.datacontract.org/2004/07/Microsoft.SharePoint"
name="detail" element="q1:SPTopologyWebServiceApplicationFault"/>
    </wsdl:message>
    <wsdl:message name="ITopologyWebServiceApplication_GetEndPoints_InputMessage">
        <wsdl:part name="parameters" element="tns:GetEndPoints"/>
    </wsdl:message>
    <wsdl:message name="ITopologyWebServiceApplication_GetEndPoints_OutputMessage">
        <wsdl:part name="parameters" element="tns:GetEndPointsResponse"/>
    </wsdl:message>
    <wsdl:message
name="ITopologyWebServiceApplication_GetEndPoints_SPTopologyWebServiceApplicationFaultFault_F
aultMessage">
        <wsdl:part xmlns:q1="http://schemas.datacontract.org/2004/07/Microsoft.SharePoint"
name="detail" element="q1:SPTopologyWebServiceApplicationFault"/>
    </wsdl:message>
</wsdl:definitions>
```

7 Appendix B: Full XML Schema

Schema name	Prefix	Section
http://schemas.datacontract.org/2004/07/Microsoft.SharePoint.Administration	tns1	7.1
http://schemas.datacontract.org/2004/07/Microsoft.SharePoint	q1	7.2
http://schemas.microsoft.com/2003/10/Serialization/Arrays	tns2	7.3
http://schemas.microsoft.com/2003/10/Serialization/	ser	7.4
http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel	tns3	7.5
http://schemas.datacontract.org/2004/07/System	q2	7.6
http://tempuri.org/	tns	7.7

For ease of implementation, the following sections provide the full XML schema for this protocol.

7.1 http://schemas.datacontract.org/2004/07/Microsoft.SharePoint.Administration Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
xmlns:tns1="http://schemas.datacontract.org/2004/07/Microsoft.SharePoint.Administration"
xmlns:ser="http://schemas.microsoft.com/2003/10/Serialization/"
elementFormDefault="qualified"
targetNamespace="http://schemas.datacontract.org/2004/07/Microsoft.SharePoint.Administration"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="ArrayOfSPSharedServiceApplicationInfo">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="SPSharedServiceApplicationInfo"
nillable="true" type="tns1:SPSharedServiceApplicationInfo"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="ArrayOfSPSharedServiceApplicationInfo" nillable="true"
type="tns1:ArrayOfSPSharedServiceApplicationInfo"/>
  <xs:complexType name="SPSharedServiceApplicationInfo">
    <xs:sequence>
      <xs:element minOccurs="0" name="ApplicationClassId" type="ser:guid"/>
      <xs:element xmlns:q2="http://schemas.datacontract.org/2004/07/System" minOccurs="0"
name="ApplicationVersion" nillable="true" type="q2:Version"/>
      <xs:element minOccurs="0" name="Comments" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="DisplayName" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="TermsOfServiceUri" nillable="true" type="xs:anyURI"/>
      <xs:element minOccurs="0" name="Uri" nillable="true" type="xs:anyURI"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="SPSharedServiceApplicationInfo" nillable="true"
type="tns1:SPSharedServiceApplicationInfo"/>
  <xs:import namespace="http://schemas.datacontract.org/2004/07/System"/>
  <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/"/>
</xs:schema>
```

7.2 http://schemas.datacontract.org/2004/07/Microsoft.SharePoint Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:q1="http://schemas.datacontract.org/2004/07/Microsoft.SharePoint"
elementFormDefault="qualified"
```

```
targetNamespace="http://schemas.datacontract.org/2004/07/Microsoft.SharePoint"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:complexType name="SPTopologyWebServiceApplicationFault">
    <xs:sequence>
      <xs:element minOccurs="0" name="FaultReason" nillable="true" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="SPTopologyWebServiceApplicationFault" nillable="true"
type="q1:SPTopologyWebServiceApplicationFault"/>
</xs:schema>
```

7.3   http://schemas.microsoft.com/2003/10/Serialization/Arrays Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:tns2="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="ArrayOfanyURI">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="anyURI" nillable="true"
type="xs:anyURI"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="ArrayOfanyURI" nillable="true" type="tns2:ArrayOfanyURI"/>
</xs:schema>
```

7.4   http://schemas.microsoft.com/2003/10/Serialization/ Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:ser="http://schemas.microsoft.com/2003/10/Serialization/"
attributeFormDefault="qualified" elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/2003/10/Serialization/"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="anyType" nillable="true" type="xs:anyType"/>
  <xs:element name="anyURI" nillable="true" type="xs:anyURI"/>
  <xs:element name="base64Binary" nillable="true" type="xs:base64Binary"/>
  <xs:element name="boolean" nillable="true" type="xs:boolean"/>
  <xs:element name="byte" nillable="true" type="xs:byte"/>
  <xs:element name="dateTime" nillable="true" type="xs:dateTime"/>
  <xs:element name="decimal" nillable="true" type="xs:decimal"/>
  <xs:element name="double" nillable="true" type="xs:double"/>
  <xs:element name="float" nillable="true" type="xs:float"/>
  <xs:element name="int" nillable="true" type="xs:int"/>
  <xs:element name="long" nillable="true" type="xs:long"/>
  <xs:element name="QName" nillable="true" type="xs:QName"/>
  <xs:element name="short" nillable="true" type="xs:short"/>
  <xs:element name="string" nillable="true" type="xs:string"/>
  <xs:element name="unsignedByte" nillable="true" type="xs:unsignedByte"/>
  <xs:element name="unsignedInt" nillable="true" type="xs:unsignedInt"/>
  <xs:element name="unsignedLong" nillable="true" type="xs:unsignedLong"/>
  <xs:element name="unsignedShort" nillable="true" type="xs:unsignedShort"/>
  <xs:element name="char" nillable="true" type="ser:char"/>
  <xs:simpleType name="char">
    <xs:restriction base="xs:int"/>
  </xs:simpleType>
  <xs:element name="duration" nillable="true" type="ser:duration"/>
  <xs:simpleType name="duration">
    <xs:restriction base="xs:duration">
      <xs:pattern value="\-?P(\d*D)?(T(\d*H)?(\d*M)?(\d*(\.\d*)?S)?)?"/>
      <xs:minInclusive value="-P10675199DT2H48M5.4775808S"/>
      <xs:maxInclusive value="P10675199DT2H48M5.4775807S"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="guid" nillable="true" type="ser:guid"/>
```

```
  <xs:simpleType name="guid">
    <xs:restriction base="xs:string">
      <xs:pattern value="[\da-fA-F]{8}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-
F]{12}"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:attribute name="FactoryType" type="xs:QName"/>
  <xs:attribute name="Id" type="xs:ID"/>
  <xs:attribute name="Ref" type="xs:IDREF"/>
</xs:schema>
```

7.5 http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
xmlns:tns3="http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel"
elementFormDefault="qualified"
targetNamespace="http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="ReadOnlyCollectionOfanyURI">
    <xs:sequence>
      <xs:element xmlns:tns2="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
name="list" nillable="true" type="tns2:ArrayOfanyURI"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="ReadOnlyCollectionOfanyURI" nillable="true"
type="tns3:ReadOnlyCollectionOfanyURI"/>
  <xs:complexType name="ReadOnlyCollectionOfSPSharedServiceApplicationInfo1RC ScQDg">
    <xs:sequence>
      <xs:element
xmlns:tns1="http://schemas.datacontract.org/2004/07/Microsoft.SharePoint.Administration"
name="list" nillable="true" type="tns1:ArrayOfSPSharedServiceApplicationInfo"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="ReadOnlyCollectionOfSPSharedServiceApplicationInfo1RC_ScQDg"
nillable="true" type="tns3:ReadOnlyCollectionOfSPSharedServiceApplicationInfo1RC ScQDg"/>
  <xs:import
namespace="http://schemas.datacontract.org/2004/07/Microsoft.SharePoint.Administration"/>
  <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
</xs:schema>
```

7.6 http://schemas.datacontract.org/2004/07/System Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:q2="http://schemas.datacontract.org/2004/07/System"
elementFormDefault="qualified"
targetNamespace="http://schemas.datacontract.org/2004/07/System"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="Version">
    <xs:sequence>
      <xs:element name=" Build" type="xs:int"/>
      <xs:element name="_Major" type="xs:int"/>
      <xs:element name="_Minor" type="xs:int"/>
      <xs:element name="_Revision" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Version" nillable="true" type="q2:Version"/>
</xs:schema>
```

7.7 http://tempuri.org/ Schema

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="GetEndPoints">
    <xs:complexType>
      <xs:sequence>
        <xs:element xmlns:ser="http://schemas.microsoft.com/2003/10/Serialization/"
minOccurs="0" name="serviceId" type="ser:guid"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="GetEndPointsResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element
xmlns:tns3="http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel"
minOccurs="0" name="GetEndPointsResult" nillable="true"
type="tns3:ReadOnlyCollectionOfanyURI"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="EnumerateSharedServiceApplications">
    <xs:complexType>
      <xs:sequence/>
    </xs:complexType>
  </xs:element>
  <xs:element name="EnumerateSharedServiceApplicationsResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element
xmlns:tns3="http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel"
minOccurs="0" name="EnumerateSharedServiceApplicationsResult" nillable="true"
type="tns3:ReadOnlyCollectionOfSPSharedServiceApplicationInfo1RC_ScQDg"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:import
namespace="http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel"/>
  <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/"/>
</xs:schema>
```

8   Appendix C: 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.

- Microsoft SharePoint Foundation 2010

- Microsoft SharePoint Foundation 2013

- Microsoft SharePoint Server 2016

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.

9 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.

- The removal of a document from the documentation set.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **Editorial** means that the formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the technical content of the document is identical to the last released version.

Major and minor changes can be described further using the following change types:

- New content added.

- Content updated.

- Content removed.

- New product behavior note added.

- Product behavior note updated.

- Product behavior note removed.

- New protocol syntax added.

- Protocol syntax updated.

- Protocol syntax removed.

- New content added due to protocol revision.

- Content updated due to protocol revision.

- Content removed due to protocol revision.

- New protocol syntax added due to protocol revision.

- Protocol syntax updated due to protocol revision.

- Protocol syntax removed due to protocol revision.

- Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated**.

Some important terms used in the change type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.

- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

The changes made to this document are listed in the following table. For more information, please contact dochelp@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
3.1.4.1.3.2 ArrayOfSPSharedServiceApplicationInfo	Clarified the description to indicate that the ArrayOfSPSharedServiceApplicationInfo complex type is a list of SPSharedServiceApplicationInfo objects.	N	Content update.

10 Index