

[MS-ECTPWPS]:

External Content Type Picker Web Service Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
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	Minor	Updated 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	1.04	None	No changes to the meaning, language, or formatting of the technical content.
9/27/2010	1.04	None	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	1.04	None	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	1.05	Minor	Clarified the meaning of the technical content.
3/18/2011	1.05	None	No changes to the meaning, language, or formatting of the technical content.
6/10/2011	1.05	None	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	1.6	Minor	Clarified the meaning of the technical content.
4/11/2012	1.6	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	1.6	None	No changes to the meaning, language, or formatting of the technical content.
9/12/2012	1.6	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	1.6.1	Editorial	Changed language and formatting in the technical content.
2/11/2013	1.6.1	None	No changes to the meaning, language, or formatting of the technical content.
7/30/2013	2.0	Major	Significantly changed the technical content.
11/18/2013	2.0	None	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	2.0	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	2.1	Minor	Clarified the meaning of the technical content.
7/31/2014	2.1	None	No changes to the meaning, language, or formatting of the

Date	Revision History	Revision Class	Comments
			technical content.
10/30/2014	2.1	None	No changes to the meaning, language, or formatting of the technical content.
3/16/2015	3.0	Major	Significantly changed the technical content.
2/26/2016	4.0	Major	Significantly changed the technical content.

Table of Contents

1	Introduction	6
1.1	Glossary	6
1.2	References	8
1.2.1	Normative References	8
1.2.2	Informative References	9
1.3	Overview	9
1.4	Relationship to Other Protocols	9
1.5	Prerequisites/Preconditions	9
1.6	Applicability Statement	9
1.7	Versioning and Capability Negotiation	9
1.8	Vendor-Extensible Fields	10
1.9	Standards Assignments.....	10
2	Messages.....	11
2.1	Transport	11
2.2	Common Message Syntax	11
2.2.1	Namespaces	11
2.2.2	Messages.....	11
2.2.3	Elements	11
2.2.4	Complex Types.....	11
2.2.4.1	ArrayOfString	12
2.2.5	Simple Types	12
2.2.6	Attributes	12
2.2.7	Groups	12
2.2.8	Attribute Groups.....	12
3	Protocol Details	13
3.1	Server Details.....	13
3.1.1	Abstract Data Model.....	13
3.1.2	Timers	13
3.1.3	Initialization.....	13
3.1.4	Message Processing Events and Sequencing Rules	14
3.1.4.1	DecodeEntityInstanceId	14
3.1.4.1.1	Messages	14
3.1.4.1.1.1	IResolverPickerService_DecomposeEntityInstanceId_InputMessage	15
3.1.4.1.1.2	IResolverPickerService_DecomposeEntityInstanceId_OutputMessage	15
3.1.4.1.2	Elements.....	15
3.1.4.1.2.1	DecodeEntityInstanceId	15
3.1.4.1.2.2	DecodeEntityInstanceIdResponse.....	18
3.1.4.1.3	Complex Types	18
3.1.4.1.4	Simple Types	18
3.1.4.1.5	Attributes	18
3.1.4.1.6	Groups.....	18
3.1.4.1.7	Attribute Groups.....	19
3.1.4.2	GetEntityInstances	19
3.1.4.2.1	Messages	20
3.1.4.2.1.1	IResolverPickerService_GetEntityInstances_InputMessage	20
3.1.4.2.1.2	IResolverPickerService_GetEntityInstances_OutputMessage	21
3.1.4.2.2	Elements.....	21
3.1.4.2.2.1	GetEntityInstances	21
3.1.4.2.2.2	GetEntityInstancesResponse	22
3.1.4.2.3	Complex Types	26
3.1.4.2.3.1	ArrayOfBoolean	26
3.1.4.2.4	Simple Types	27
3.1.4.2.5	Attributes	27

3.1.4.2.6	Groups.....	27
3.1.4.2.7	Attribute Groups.....	27
3.1.4.3	ReadEntityInstance	27
3.1.4.3.1	Messages	28
3.1.4.3.1.1	IResolverPickerService_ReadEntityInstance_InputMessage	28
3.1.4.3.1.2	IResolverPickerService_ReadEntityInstance_OutputMessage	28
3.1.4.3.2	Elements.....	28
3.1.4.3.2.1	ReadEntityInstance	28
3.1.4.3.2.2	ReadEntityInstanceResponse.....	29
3.1.4.3.3	Complex Types	30
3.1.4.3.4	Simple Types	30
3.1.4.3.5	Attributes	30
3.1.4.3.6	Groups.....	30
3.1.4.3.7	Attribute Groups.....	30
3.1.5	Timer Events.....	30
3.1.6	Other Local Events.....	30
4	Protocol Examples	31
4.1	Getting Entity Instances.....	31
4.2	Reading an Entity Instance.....	32
5	Security	33
5.1	Security Considerations for Implementers	33
5.2	Index of Security Parameters	33
6	Appendix A: Full WSDL	34
7	Appendix B: Full XML Schema.....	36
8	Appendix C: Product Behavior	38
9	Change Tracking.....	39
10	Index.....	41

1 Introduction

The External Content Type Picker Web Service Protocol enables protocol clients to find and resolve external data on a protocol server.

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:

ASCII: The American Standard Code for Information Interchange (ASCII) is an 8-bit character-encoding scheme based on the English alphabet. ASCII codes represent text in computers, communications equipment, and other devices that work with text. ASCII refers to a single 8-bit ASCII character or an array of 8-bit ASCII characters with the high bit of each character set to zero.

authentication: The act of proving an identity to a server while providing key material that binds the identity to subsequent communications.

base64 encoding: A binary-to-text encoding scheme whereby an arbitrary sequence of bytes is converted to a sequence of printable **ASCII** characters, as described in [\[RFC4648\]](#).

common language runtime (CLR): The core runtime engine in the Microsoft .NET Framework for executing applications. The common language runtime supplies managed code with services such as cross-language integration, code access security, object lifetime management, and debugging and profiling support.

Entity: A type of DataClass that represents a type of business data object that is stored in a line-of-business (LOB) system and whose instances have a persistent EntityInstanceId.

EntityInstance: A set of **Field (2)** values that have a unique identity that represents a specific instance of an Entity, and are stored in a line-of-business (LOB) system.

field: (1) An element or attribute (1) in a data source that can contain data.

(2) The data elements that constitute an **Entity** in a line-of-business (LOB) system.

Finder: A type of MethodInstance that can be called to return a collection of zero or more EntityInstances of a specific Entity. Finder input is defined by the FilterDescriptors that are contained in the Method that contains the Finder.

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.

Hypertext Transfer Protocol Secure (HTTPS): An extension of HTTP that securely encrypts and decrypts web page requests. In some older protocols, "Hypertext Transfer Protocol over Secure Sockets Layer" is still used (Secure Sockets Layer has been deprecated). For more information, see [\[SSL3\]](#) and [\[RFC5246\]](#).

Identifier: The **Field (2)** or **Fields (2)** that define the Identity of an EntityInstance. Also referred to as Key.

LobSystem: A type of **MetadataObject** that represents a specific version of a line-of business (LOB) system. An LOB system can be a database or a web service.

LobSystemInstance: A type of **MetadataObject** that represents a specific deployed instance of a line-of-business (LOB) system, as represented by a **LobSystem**. LobSystemInstances are contained by LobSystems. LobSystemInstance Properties describe how to connect to an instance of the LobSystem that contains them by providing information such as the server name, connection string, and authentication mode.

metadata store: A database that is stored on a back-end database server and contains all stored procedures and storage for the **MetadataObject** types.

MetadataObject: An abstract data structure that consists of a set of attributes (1) that represent a LobSystem, LobSystemInstance, DataClass, Entity, Method, MethodInstance, Parameter, TypeDescriptor, Identifier, FilterDescriptor, Action, ActionParameter, or Association.

site: A group of related webpages that is hosted by a server on the World Wide Web or an intranet. Each website has its own entry points, metadata, administration settings, and workflows. Also referred to as web site.

SOAP: A lightweight protocol for exchanging structured information in a decentralized, distributed environment. **SOAP** uses XML technologies to define an extensible messaging framework, which provides a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular programming model and other implementation-specific semantics. SOAP 1.2 supersedes SOAP 1.1. See [\[SOAP1.2-1/2003\]](#).

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.

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.

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.

[ISO-8601] International Organization for Standardization, "Data Elements and Interchange Formats - Information Interchange - Representation of Dates and Times", ISO/IEC 8601:2004, December 2004, <http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=40874&ICS1=1&ICS2=140&ICS3=30>

Note There is a charge to download the specification.

[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>

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

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

[WSA1.0] World Wide Web Consortium, "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>

[XML10] World Wide Web Consortium, "Extensible Markup Language (XML) 1.0 (Third Edition)", February 2004, <http://www.w3.org/TR/2004/REC-xml-20040204/>

[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 Overview

This protocol enables a protocol client to find and resolve **EntityInstances**. The protocol allows the protocol client to pass criteria, which consist of a set of values describing an **Entity** and a search query, to the protocol server to receive back a list of EntityInstances that match the criteria. The protocol also allows the protocol client to pass the value of an identifier of an EntityInstance to the protocol server to receive back a set of values describing the identifier or describing the instance.

A typical scenario for using this protocol is an application that allows users to retrieve a set of EntityInstances matching a query or to verify if user input can match the key of an existing EntityInstance and show detailed data about that EntityInstance.

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 **HTTP**, as described in [RFC2616], or **Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)**, as described in [RFC2818].

The following diagram shows the underlying messaging and transport stack used by the protocol:

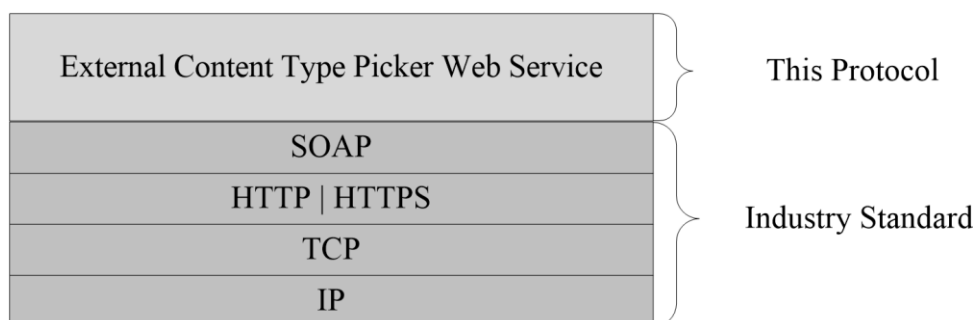


Figure 1: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

This protocol operates against a **site** that is identified by a **Uniform Resource Locator (URL)** that is known by protocol clients. The protocol server endpoint is formed by appending `"/_vti_bin/BDCResolverPickerService.svc"` to the URL of the site, for example: `http://www.contoso.com/Repository/_vti_bin/BDCResolverPickerService.svc`.

This protocol assumes that **authentication** has been performed by the underlying protocols.

1.6 Applicability Statement

None.

1.7 Versioning and Capability Negotiation

This protocol uses multiple transports with SOAP as specified in section [2.1](#).

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

Protocol servers MUST support SOAP over HTTP. Protocol servers SHOULD additionally support SOAP over HTTPS for securing communication with protocol clients.

Protocol messages MUST be formatted as specified in [\[SOAP1.1\]](#) section 4. Protocol server faults MUST be returned either using HTTP status codes as specified in [\[RFC2616\]](#) section 10 or using **SOAP faults** as specified in [\[SOAP1.1\]](#) section 4.4.

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
soap	http://schemas.xmlsoap.org/wsdl/soap/	[SOAP1.1]
tns	http://tempuri.org/	
tns1	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
ArrayOfString	The ArrayOfString complex type represents a list of xs:string as specified in [XMLSCHEMA2] .

2.2.4.1 ArrayOfString

Namespace: http://tempuri.org/

The **ArrayOfString** complex type represents a list of **xs:string** as specified in [\[XMLSCHEMA2\]](#).

This complex type is defined as follows.

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

string: An element in the list.

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

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.

3.1 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 MUST maintain lists of following **MetadataObject** types: **LobSystem**, **LobSystemInstance**, **Entity**, **Identifiers**, **Finder** and **Field**. The protocol server MUST maintain the following relationships among these **MetadataObject** types:

- Each **Entity** is contained by a **LobSystem**.
- Each **LobSystemInstance** is contained by a **LobSystem**.
- Each **Field** is contained by either an **Entity** or another **Field**.
- Each **Finder** is contained by an **Entity**.
- Each **Entity** contains 0 or 1 **DefaultFinders**.
- Each **Identifier** MUST be contained by an **Entity**.

The protocol server MUST follow these rules:

- All **Entities** that the server maintains MUST have unique names.
- All **LobSystemInstances** contained in one **LobSystem** MUST have unique names.
- All **Finders** contained in one **Entity** MUST have unique names.
- Exactly one of the **Finders** of an **Entity** MUST be identified as the **DefaultFinder** for that **Entity**.
- All **Fields** contained in one **Entity** MUST have unique names.
- All **Fields** contained within the same parent **Field** MUST have unique names.
- All **Identifiers** within an **Entity** MUST have a unique positive integer associated with them, valued between 1 and the number of **Identifiers**.
- All **LobSystems** contained by a **metadata store** MUST have unique names.

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
DecodeEntityInstanceId	This operation is used to retrieve the data values of an EntityInstance's Identifiers encoded in the input.
GetEntityInstances	This operation is used to retrieve the data values of a list of Entity Fields associated with one or more EntityInstances matching the input criteria.
ReadEntityInstance	This operation is used to retrieve the data values of the EntityInstance Identifiers and the display name value of the EntityInstance identified by the input criteria.

3.1.4.1 DecodeEntityInstanceId

This operation is used to retrieve the data values of an EntityInstance's Identifiers encoded in the input.

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

```
<wsdl:operation name="DecodeEntityInstanceId" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input wsaw:Action="http://tempuri.org/IResolverPickerService/DecodeEntityInstanceId"
    message="tns:IResolverPickerService_DecodeEntityInstanceId_InputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output
    wsaw:Action="http://tempuri.org/IResolverPickerService/DecodeEntityInstanceIdResponse"
    message="tns:IResolverPickerService_DecodeEntityInstanceId_OutputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>
```

The protocol client sends an **IResolverPickerService_DecodeEntityInstanceId_InputMessage** request message, and the protocol server responds with an **IResolverPickerService_DecodeEntityInstanceId_OutputMessage** response message, as follows:

1. The caller of this operation MUST provide the encoded data values of the **Identifiers** which are to be decoded.
2. If the provided encoded data values of the **Identifiers** can be decoded, the protocol server MUST return the list of the values of the **EntityInstance's Identifiers** encoded in the input and MUST return the success status set to **true**.

This operation MUST return a **DecodeEntityInstanceIdResponse** (section [3.1.4.1.2.2](#)).

If any unexpected error condition is met, the protocol server MUST respond with a SOAP fault, which MUST comply with the SOAP fault specification in section 4.4 of [\[SOAP1.1\]](#). The fault code MUST be set to "InternalServiceFault". The fault string MUST be set to an implementation-specific string indicating that an error has occurred.

3.1.4.1.1 Messages

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

Message	Description
IResolverPickerService_DecodeEntityInstanceId_InputMessage	The request WSDL message for the DecodeEntityInstanceId WSDL operation.
IResolverPickerService_DecodeEntityInstanceId_OutputMessage	The response WSDL message for the DecodeEntityInstanceId WSDL operation.

3.1.4.1.1.1 IResolverPickerService_DecodeEntityInstanceId_InputMessage

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

The **SOAP action** value is:

```
http://tempuri.org/IResolverPickerService/DecodeEntityInstanceId
```

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

3.1.4.1.1.2 IResolverPickerService_DecodeEntityInstanceId_OutputMessage

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

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

3.1.4.1.2 Elements

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

Element	Description
DecodeEntityInstanceId	The input data for the DecodeEntityInstanceId WSDL operation.
DecodeEntityInstanceIdResponse	The result data for the DecodeEntityInstanceId WSDL operation.

3.1.4.1.2.1 DecodeEntityInstanceId

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

```
<xs:element name="DecodeEntityInstanceId" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="bstrSiteId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="bstrEntityInstanceId" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="fFormatAsXml" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

bstrSiteId: MUST be ignored by the protocol server.

bstrEntityInstanceId: the value of the serialized reference to the EntityInstance. The value MUST be present. The format of the reference to the **EntityInstance** MUST be as follows:

- The string MUST begin with the serialized length of the Entity namespace.
- The string MUST then contain a colon (:).
- The string MUST then contain the **Entity** namespace.
- The string MUST then contain the serialized length of the **Entity** name.
- The string MUST then contain a colon (:).
- The string MUST then contain the **Entity** name.
- The string MUST then contain the serialized length of the name of a Finder contained in the **Entity**.
- The string MUST then contain a colon (:).
- The string MUST then contain the Finder name.
- The string MUST then contain the serialized length of the name of a LobSystemInstance contained in the same LobSystem as the **Entity** identified by the **Entity** namespace and **Entity** name.
- The string MUST then contain a colon (:).
- The string MUST then contain the LobSystemInstance name.
- The rest of the reference to the **EntityInstance** MUST consist of a serialization of the values and types of the data values of the **EntityInstance Identifiers**:
 - The serialization of the data values of the **EntityInstance Identifiers** MUST be composed of the concatenation of the encodings of the data value of each **EntityInstance Identifier** and type.
 - Each data value of an **EntityInstance Identifier** MUST be encoded as follows:
 - "A", if the **common language runtime (CLR)** type of the data value of the **EntityInstance Identifier** is "System.Boolean", and the value is equal to **true**.
 - "a", if the CLR type of the data value of the **EntityInstance Identifier** is "System.Boolean", and the value is equal to **false**.
 - "b", if the CLR type of the data value of the **EntityInstance Identifier** is "System.Byte", followed by the **base64 encoding** of the value. The base64 encoding in this and subsequent descriptions of the data value of the **EntityInstance Identifier** encoding MUST be performed according to [\[RFC4648\]](#).
 - "C", if the CLR type of the data value of the **EntityInstance Identifier** is "System.Char", followed by the base64 encoding of the **Char** value.
 - "D", if the CLR type of the data value of the **EntityInstance Identifier** is **System.DateTime**, followed by a character representing the CLR **DateTimeKind** of the data value and then followed by the base64 encoding of the binary representation of the **DateTime** value.
 - The character representing the **DateTimeKind** of the data value MUST be obtained as follows:

- "a", if the CLR **DateTimeKind** is "Unspecified".
- "b", if the CLR **DateTimeKind** is "UTC".
- "c", if the CLR **DateTimeKind** is "Local".
- The binary representation of the **DateTime** value MUST be obtained as follows:
 - Obtain the **Ticks** representation of the **DateTime** value, as the number of 100-nanosecond intervals that have elapsed since 12:00:00 midnight, January 1, 0001.
 - If the **DateTime** is not local, its binary representation MUST be equal to the **Ticks** value obtained in the previous step.
 - If the **DateTime** is local, obtain the UTC equivalent **Ticks** representation by adjusting for the **TimeZone** of the **DateTime**.
 - If the number of **Ticks** is negative, add 4611686018427387904.
 - The binary representation MUST be the result of the bitwise OR between the resulting number of **Ticks** and -9223372036854775808.
- "E", if the CLR type of the data value of the **EntityInstance Identifier** is **System.Decimal**, followed by the base64 encoding of the length of the base64 encoding of the value, followed by the base64 encoding of the string representation of the value.
- "F", if the CLR type of the data value of the **EntityInstance Identifier** is "System.Double", followed by the base64 encoding of the value.
- "G", if the CLR type of the data value of the **EntityInstance Identifier** is "System.Guid", followed by the base64 encoding of the string representation of the value.
- "H", if the CLR type of the data value of the **EntityInstance Identifier** is "System.Int16", followed by the base64 encoding of the value.
- "I", if the CLR type of the data value of the **EntityInstance Identifier** is "System.Int32", followed by the base64 encoding of the value.
- "I", if the CLR type of the data value of the **EntityInstance Identifier** is "System.Int64", followed by the base64 encoding of the value.
- "h", if the CLR type of the data value of the **EntityInstance Identifier** is "System.SByte", followed by the base64 encoding of the arithmetic sum between the value and the constant value 128.
- "f", if the CLR type of the data value of the **EntityInstance Identifier** is "System.Single", followed by the base64 encoding of the value.
- "S", if the CLR type of the data value of the **EntityInstance Identifier** is "System.String", followed by the base64 encoding of the length of the base64 encoding of the value, followed by the base64 encoding of the value.
- "d", if the CLR type of the data value of the **EntityInstance Identifier** is "System.TimeSpan", followed by the base64 encoding of the **Ticks** representation of the value. The **Ticks** representation of the value MUST be the number of 100 nanosecond intervals that are equivalent in duration to the value.
- "B", if the CLR type of the data value of the **EntityInstance Identifier** is "System.UInt16", followed by the base64 encoding of the value.

- "u", if the CLR type of the data value of the **EntityInstance Identifier** is "System.UInt32", followed by the base64 encoding of the value.
- "U", if the CLR type of the data value of the **EntityInstance Identifier** is "System.UInt64", followed by the base64 encoding of the value.

fFormatAsXml: Flag indicating whether returned **EntityInstance Identifiers** of CLR type System.DateTime, if any, need to be formatted according to CLR serialization rules or according to [\[ISO-8601\]](#). The value MUST be present.

- If **fFormatAsXml** is **true**, any returned **EntityInstance Identifiers** of CLR type "System.DateTime" MUST be represented according to the extended format described in [\[ISO-8601\]](#).
- If **fFormatAsXml** is **false**, any returned **EntityInstance Identifiers** of CLR type "System.DateTime" MUST be encoded as the string representation of the **Ticks** represented by the data value of the **EntityInstance Identifier**. The **Ticks** represented by a value of CLR type "System.DateTime" MUST be the number of 100-nanosecond intervals that have elapsed since 12:00:00 midnight, January 1, 0001.

3.1.4.1.2.2 DecodeEntityInstanceIdResponse

The **DecodeEntityInstanceIdResponse** element specifies the result data for the **DecodeEntityInstanceId** WSDL operation.

```
<xs:element name="DecodeEntityInstanceIdResponse"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="DecodeEntityInstanceIdResult"
        type="tns:ArrayOfString"/>
      <xs:element minOccurs="0" maxOccurs="1" name="message" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="success" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

DecodeEntityInstanceIdResult: The list of data values of the **EntityInstance Identifiers** obtained by decoding the provided input. Each value MUST be the CLR-provided serialized representation of the corresponding data value of the **EntityInstance Identifier**. Each value MUST be encoded as specified in [\[XML10\]](#).

message: Contains errors or warnings resulting from the process of decoding the provided input, if any. The message MUST be encoded as specified in [\[XML10\]](#).

success: A flag indicating whether the operation finished without any unexpected errors. The value MUST [<1>](#) be set to true if the operation finished without any unexpected errors, false otherwise. The value MUST be ignored by the protocol client. The value MUST be present.

3.1.4.1.3 Complex Types

None.

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 GetEntityInstances

This operation is used to retrieve the data values of a list of Entity Fields associated with one or more EntityInstances matching the input criteria.

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

```
<wsdl:operation name="GetEntityInstances" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input wsaw:Action="http://tempuri.org/IResolverPickerService/GetEntityInstances"
    message="tns:IResolverPickerService_GetEntityInstances_InputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output
    wsaw:Action="http://tempuri.org/IResolverPickerService/GetEntityInstancesResponse"
    message="tns:IResolverPickerService_GetEntityInstances_OutputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>
```

This operation is used to search for **EntityInstances** matching the input criteria. The input criteria are the LobSystemInstance name, the **Entity** name, the **Entity** namespace, the **Entity** Finder name, the name of the **Field** used to visualize the **EntityInstance** in a UI, the user-provided search token, whether the method MUST resolve or MUST search for **EntityInstances** and the maximum number of results to be returned by the protocol server.

The protocol client sends an **IResolverPickerService_GetEntityInstances_InputMessage** request message, and the protocol server responds with an **IResolverPickerService_GetEntityInstances_OutputMessage** response message, as follows:

- The caller of this operation MUST specify:
 - The name of the LobSystemInstance which is to be searched.
 - The name of the **Entity** which is to be searched.
 - The **Entity** namespace which is to be searched
 - The name of the **Field** within the **EntityInstance** that contains the suggested display text for visualizing the **EntityInstance** in the UI presented to the user by the protocol client.
 - The value to search for, representing the input from the user that is to be used as the search string.
 - Whether this operation MUST attempt to search for **EntityInstances** or MUST resolve the **EntityInstance**.
 - The maximum number of **EntityInstances** to return.
- In addition, the caller of this operation:

- MUST specify the name of the Finder to be used, if the protocol server MUST use a particular Finder to identify the appropriate **EntityInstances**. The caller MUST NOT provide the name of a Finder if the protocol server MUST use an implementation-specific Finder to identify the appropriate **EntityInstances**.
- MUST specify the name of the **Field**, if the protocol server MUST use a particular **Field** to identify the value used for the visualization of the **EntityInstances** in the UI presented to the user by the protocol client. The caller MUST NOT provide the name of a **Field** if the protocol server MUST use an implementation-specific **Field** to identify the value used for the visualization of the **EntityInstances** in the UI presented to the user by the protocol client.
- If at least one **EntityInstance** is found, the protocol server MUST return:
 - The matching **EntityInstances'** list of name-value pairs representing the **Entity Field** name and the corresponding **EntityInstance Field** value.
 - The list of hints indicating whether to display a specific name-value pair in the UI presented to the user by the protocol client.
 - The success flag, set to true if the operation finished without any unexpected errors, false otherwise.
- If no **EntityInstances** are found matching the input criteria, the protocol server MUST return:
 - An empty list of name-value pairs.
 - An empty list of hints indicating whether to display a specific name-value pair in the UI presented to the user by the protocol client.
 - The success flag, set to true if the operation finished without any unexpected errors, false otherwise.
- In addition, for any expected errors or warnings encountered during the operation, the protocol server MUST return a description of the error or warning in the Message. As an example, an error message will be returned if one of the parameters, such as the **Entity** namespace, is incorrect.

This operation MUST return **GetEntityInstanceResponse** (section [3.1.4.2.2.2](#)).

If any unexpected error condition is met, the protocol server MUST respond with a SOAP fault, which MUST comply with the SOAP fault specification in section 4.4 of [\[SOAP1.1\]](#). The faultcode MUST be set to **InternalServiceFault**. The **faultstring** MUST be set to an implementation-specific string indicating that an error has occurred.

3.1.4.2.1 Messages

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

Message	Description
IResolverPickerService_GetEntityInstances_InputMessage	The request WSDL message for the GetEntityInstances WSDL operation.
IResolverPickerService_GetEntityInstances_OutputMessage	The response WSDL message for the GetEntityInstances WSDL operation.

3.1.4.2.1.1 IResolverPickerService_GetEntityInstances_InputMessage

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

The SOAP action value is:

```
http://tempuri.org/IResolverPickerService/GetEntityInstances
```

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

3.1.4.2.1.2 IResolverPickerService_GetEntityInstances_OutputMessage

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

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

3.1.4.2.2 Elements

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

Element	Description
GetEntityInstances	The input data for the GetEntityInstances WSDL operation.
GetEntityInstancesResponse	The result data for the GetEntityInstances WSDL operation.

3.1.4.2.2.1 GetEntityInstances

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

```
<xs:element name="GetEntityInstances" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="siteId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="systemInstanceName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="entityNamespace" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="entityName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="finderName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="displayFieldName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="searchToken" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="usedForPicking" type="xs:boolean"/>
      <xs:element minOccurs="1" maxOccurs="1" name="maxResults" type="xs:unsignedInt"/>
      <xs:element minOccurs="1" maxOccurs="1" name="refreshInterval" type="xs:unsignedInt"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

siteId: MUST be ignored by the protocol server.

systemInstanceName: The name of the LobSystemInstance to be searched. The value MUST be present. The value MUST be encoded as specified in [\[XML10\]](#).

entityNamespace: The Entity namespace to be searched. The value MUST be present. The value MUST be encoded as specified in [\[XML10\]](#).

entityName: The name of the **Entity** to be searched. The value MUST be present. The value MUST be encoded as specified in [\[XML10\]](#).

finderName: The name of the **Entity** Finder to be used for searching. If the Finder name is not specified, the default Finder of the **Entity** MUST be used. The value MUST be encoded as specified in [XML10].

displayFieldName: The location, within the EntityInstance, from where to obtain the suggested value to be used when displaying the **EntityInstance** in the UI presented to the user by the protocol client. The location MUST be specified in a hierarchical fashion. The first token in the location MUST be the name of one of the Fields of the **Entity**. Any subsequent token MUST be the name of a **Field** contained in the **Field** identified by the previous token. Tokens MUST be separated by a single period (.). If appearing within the name of a token, the following characters MUST be replaced by their encoded string counterparts:

Character	Encoded string counterpart
Opening square bracket ([)	Backslash, opening square bracket (\[)
Period (.)	Backslash, period (\.)
Backslash (\)	Backslash, backslash (\\)

The entire **displayFieldName** MUST be encoded as specified in [XML10].

If a **displayFieldName** is not specified, an implementation-specific **Field** of the **Entity** MUST be used.

searchToken: The user-provided string used to identify the **EntityInstances**. The value MUST be present. The value MUST be encoded as specified in [XML10]

usedForPicking: Flag indicating whether this operation MUST attempt to resolve or MUST attempt to search for **EntityInstances**. The value MUST be set to **true** if the operation is used to search for **EntityInstances**; **false** otherwise.

maxResults: The maximum number of **EntityInstances** to be returned. The value MUST be present. If more **EntityInstances** are found, the return message MUST indicate so.

refreshInterval: MUST be ignored by the protocol server. This value MUST be present.

3.1.4.2.2.2 GetEntityInstancesResponse

The **GetEntityInstancesResponse** element specifies the result data for the **GetEntityInstances** WSDL operation.

```
<xs:element name="GetEntityInstancesResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="GetEntityInstancesResult"
type="xs:unsignedInt"/>
      <xs:element minOccurs="0" maxOccurs="1" name="columnNames" type="tns:ArrayOfString"/>
      <xs:element minOccurs="0" maxOccurs="1" name="localizedColumnNames"
type="tns:ArrayOfString"/>
      <xs:element minOccurs="0" maxOccurs="1" name="showInPicker" type="tns:ArrayOfBoolean"/>
      <xs:element minOccurs="0" maxOccurs="1" name="values" type="tns:ArrayOfString"/>
      <xs:element minOccurs="1" maxOccurs="1" name="hasEntityMetadata" type="xs:boolean"/>
      <xs:element minOccurs="0" maxOccurs="1" name="message" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="success" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

GetEntityInstancesResult: The number of **EntityInstances** returned.

columnNames: an ordered list of strings indicating the names of the Fields returned for the identified Entity. The first string MUST be set to "___identities". The second string MUST be set to "___entityInstanceReference". The third string MUST be set to "___displayName". The remaining strings MUST be the names of the Fields of the identified Entity.

localizedColumnNames: an ordered list of strings indicating the locale-specific names of the Fields returned for the identified Entity. The first, second and third strings MUST be ignored by the protocol client. The remaining strings MUST be the locale-specific names of the Fields of the identified Entity. The protocol server MUST use an implementation-specific algorithm to determine the locale to which the **localizedColumnNames** will conform.

showInPicker: an ordered list of Boolean flags. The list MUST contain the same number of elements as the columnNames list. The first, second and third flags in this list MUST be ignored by the protocol client. Each of the remaining flags MUST indicate whether the **Entity Field**, identified by the name in the corresponding position in the columnNames list, is to be shown in the UI presented to the user by the protocol client. The value MUST be set to **true** if the corresponding field (2) is to be shown in the UI presented to the user by the protocol client; otherwise, the value is **false**.

values: an ordered list of string values. The list MUST contain a number of elements equal to the length of the columnNames multiplied by the value of GetEntityInstancesResult. The values list MUST be a concatenation of GetEntityInstanceResult sublists, where each of these sub lists MUST correspond to one of the identified EntityInstances and MUST contain a number of elements equal to the length of the columnNames list. Each of the elements of the values list MUST be encoded according to [\[XML10\]](#).

Each sub list MUST have the following structure:

The value of the first element of the sub list MUST be the encoded data values of the EntityInstance Identifiers corresponding to the EntityInstance. The format of the encoded data values of the EntityInstance Identifiers MUST be the encoded data values of the EntityInstance Identifiers. The data values of the EntityInstance Identifiers MUST be encoded according to the following scheme:

- The first character MUST be an underscore (_).
- The second character MUST be an underscore (_).
- The next character MUST be in the range "b" through "z". This character MUST correspond to the number of data values of EntityInstance Identifiers that are encoded in this input. If the number of encoded data values of EntityInstance Identifiers is one, this character MUST be "b". If the number of encoded data values of EntityInstance Identifiers is two, this character MUST be "c". Larger numbers of encoded data values of EntityInstance Identifiers MUST be similarly encoded in an upwards increasing manner, with "z" corresponding to 25 encoded data values of EntityInstance Identifiers. The number of encoded data values of EntityInstance Identifiers MUST be less than or equal to 25.
- The remaining characters MUST be encoded as follows:
 - The list of characters MUST be the concatenation of the encodings of each individual data value of the EntityInstance Identifier. Each individual data value of an EntityInstance Identifier MUST be encoded as a series of characters, in the manner described in the following.
 - If the data value of the EntityInstance Identifier is NULL, the first character MUST be 'p'. Otherwise, the first character MUST indicate the CLR type of the data value of the EntityInstance Identifier. The CLR type of the data value of the EntityInstance Identifier MUST be equal to one of the values in the first column of the following table. The first character MUST be the character in the same row as the CLR type of the data value of the **EntityInstance Identifier**.

CLR type of the data value of the EntityInstance Identifier	Corresponding encoding character
System.Boolean	a
System.Byte	b
System.DateTime	c
System.Decimal	d
System.Double	e
System.Int16	f
System.Int32	g
System.Int64	h
System.SByte	i
System.Single	j
System.String	k
System.UInt16	l
System.UInt32	m
System.UInt64	n
System.Guid	o
System.Char	p
System.TimeSpan	q

- The remaining characters of this encoded data value of the EntityInstance Identifier MUST represent a hexadecimal-encoded representation of the serialized representation of each data value of the EntityInstance Identifier, as follows:
 - The serialized representation of the data value of the EntityInstance Identifier MUST be represented as follows:
 - If the data value of the EntityInstance Identifier is NULL, the serialized representation MUST be "null"
 - If the CLR type of the data value of the EntityInstance Identifier is **System.DateTime**, the serialized representation MUST be the string representation of the Ticks represented by the data value of the EntityInstance Identifier. The **Ticks** represented by a value of CLR type **System.DateTime** MUST be the number of 100-nanosecond intervals that have elapsed since 12:00:00 midnight, January 1, 0001.
 - If the data value of the **EntityInstance Identifier** is not NULL and not of CLR type **System.DateTime**, the serialized representation MUST be the CLR-provided serialization of the data value of the **EntityInstance Identifier**.
 - The serialized representation of the data value of the **EntityInstance Identifier** obtained MUST be hexadecimal-encoded as follows:
 - If the CLR type of the data value of the **EntityInstance Identifier** is "System.Guid", the serialized representation MUST be encoded as follows:

- The first four characters MUST be the hexadecimal encoding of the integer that represents the string length of the serialized representation of the data value of the **EntityInstance Identifier**. The integer MUST be hexadecimal-encoded according to the same rules as if it were a character (see the following).
- The remaining characters MUST be the string representation of the data value of the **EntityInstance Identifier**.
- If the CLR type of the data value of the **EntityInstance Identifier** is not "System.Guid", the serialization representation MUST be hexadecimal-encoded as follows:
 - The first four characters MUST be the hexadecimal encoding of the integer that represents the string length of the serialized representation of the data value of the **EntityInstance Identifier**, multiplied by 4.
 - The remaining characters MUST be the hexadecimal encoding of the serialization. Each group of four characters MUST represent the hexadecimal encoding of a single character in the serialized representation. Each single character MUST be hexadecimal-encoded as detailed in what follows.
- The hexadecimal encoding of a single character MUST be represented as follows:
 - The character MUST be represented by four characters in the hexadecimal encoded representation of the **ASCII** value of the character. These characters MUST be obtained as indicated here.
 - The first character MUST correspond to the lowest-order four bits in the two-byte representation of the unencoded character. The correspondence MUST be obtained according to the following table.
 - The second character MUST correspond to the second-lowest-order four bits in the two-byte representation of the unencoded character. The correspondence MUST be obtained according to the following table.
 - The third character MUST correspond to the third-lowest-order four bits in the two-byte representation of the unencoded character. The correspondence MUST be obtained according to the following table.
 - The fourth character MUST correspond to the fourth lowest-order (that is, the highest-order) four bits in the two-byte representation of the unencoded character. The correspondence MUST be obtained according to the following table.

Value to be encoded	Corresponding encoding character
0	0
1	1
2	2
3	3
4	4
5	5
6	6

Value to be encoded	Corresponding encoding character
7	7
8	8
9	9
10	a
11	b
12	c
13	d
14	e
15	f

- As an example, if the input to be specified consists of two data values corresponding to two **EntityInstance Identifiers**, first of CLR type System.Int32 and value 4, second of CLR type System.String, with a value of "ab", the corresponding encoded **EntityInstance Identifiers** would have the following value: __cg40004300k800016002600

The value of the second element of the sub list MUST be the serialized reference to the **EntityInstance**. The format of the reference to the **EntityInstance** MUST be the same as that of the bstrEntityInstanceId input to the DecodeEntityInstanceId operation.

The value of the third element of the sub list MUST be the display name value identified for the **EntityInstance**.

The values of the remaining elements of the sub list MUST be the serialized values of the **Fields** of the **EntityInstance**.

hasEntityMetadata: If the specified Entity can be found in the metadata store, the protocol server MUST set this value to **true**. Otherwise, the protocol server MUST set this value to false. The value MUST be present.

message: Contains errors or warnings resulting from the process of resolving or searching for the **EntityInstances** according to the provided input, if any. The message MUST be encoded as specified in [XML10].

success: **true** if the operation finished without any unexpected errors; **false** otherwise. This value MUST be present.

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
ArrayOfBoolean	An ordered list of Boolean values.

3.1.4.2.3.1 ArrayOfBoolean

Namespace: http://tempuri.org/

An ordered list of Boolean values.

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

boolean: A Boolean value.

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.

3.1.4.3 ReadEntityInstance

This operation is used to retrieve the data values of the EntityInstance Identifiers and the display name value of the **EntityInstance** identified by the input criteria.

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

```
<wsdl:operation name="ReadEntityInstance" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input wsaw:Action="http://tempuri.org/IResolverPickerService/ReadEntityInstance"
    message="tns:IResolverPickerService_ReadEntityInstance_InputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output
    wsaw:Action="http://tempuri.org/IResolverPickerService/ReadEntityInstanceResponse"
    message="tns:IResolverPickerService_ReadEntityInstance_OutputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>
```

The input criteria for this operation are the reference to the EntityInstance and the name of display Field.

The protocol client sends an **IResolverPickerService_ReadEntityInstance_InputMessage** request message and the protocol server responds with an **IResolverPickerService_ReadEntityInstance_OutputMessage** response message, as follows:

- The caller of this operation MUST specify:
 - The reference to the **EntityInstance** which is to be retrieved.
 - The name of a **Field** of the **EntityInstance** whose value is to be retrieved as the display name value.

- If the referenced **EntityInstance** is found, the protocol server MUST return the data values of the **EntityInstance** Identifiers corresponding to the identified **EntityInstance** and MUST return the success status set to true.
- If the referenced **EntityInstance** is not found, the protocol server MUST return an empty array of data values of the **EntityInstance Identifiers** and the success status set to true

This operation MUST return **ReadEntityInstanceResponse** (section [3.1.4.3.2.2](#)).

If any unexpected error condition is met, the protocol server MUST respond with a SOAP fault, which MUST comply with the SOAP fault specification in section 4.4 of [\[SOAP1.1\]](#). The **faultcode** MUST be set to **InternalServiceFault**. The **faultstring** MUST be set to an implementation specific string indicating that an error has occurred.

3.1.4.3.1 Messages

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

Message	Description
IResolverPickerService_ReadEntityInstance_InputMessage	The request WSDL message for the ReadEntityInstance WSDL operation.
IResolverPickerService_ReadEntityInstance_OutputMessage	The response WSDL message for the ReadEntityInstance WSDL operation.

3.1.4.3.1.1 IResolverPickerService_ReadEntityInstance_InputMessage

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

The SOAP action value is:

```
http://tempuri.org/IResolverPickerService/ReadEntityInstance
```

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

3.1.4.3.1.2 IResolverPickerService_ReadEntityInstance_OutputMessage

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

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

3.1.4.3.2 Elements

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

Element	Description
ReadEntityInstance	The input data for the ReadEntityInstance WSDL operation.
ReadEntityInstanceResponse	The result data for the ReadEntityInstance WSDL operation.

3.1.4.3.2.1 ReadEntityInstance

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

```
<xs:element name="ReadEntityInstance" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="siteId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="entityInstanceReference"
type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="displayFieldName" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="fFormatAsXml" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

siteId: MUST be ignored by the protocol server.

entityInstanceReference: The serialized reference to the **EntityInstance**. The value MUST be present. The format of the reference to the **EntityInstance** MUST be the same as that of the **bstrEntityInstanceId** input to the **DecodeEntityInstanceId** operation.

displayFieldName: The location, within the **EntityInstance**, from where to obtain the suggested value to be used when displaying the **EntityInstance** in the UI presented to the user by the protocol client. The value MUST be specified in the format detailed in the description of the **displayFieldName** element of **GetEntityInstances** (section [3.1.4.2.2.1](#)).

fFormatAsXml: Flag indicating whether returned **EntityInstance** Identifiers of CLR type "System.DateTime", if any, need to be formatted according to CLR serialization rules or according to [\[ISO-8601\]](#). The value MUST be present.

- If **fFormatAsXml** is **true**, any returned **EntityInstance Identifiers** of CLR type "System.DateTime" MUST be represented according to the extended format described in [\[ISO-8601\]](#).
- If **fFormatAsXml** is **false**, any returned **EntityInstance Identifiers** of CLR type "System.DateTime" MUST be encoded as the string representation of the **Ticks** represented by the data value of the **EntityInstance Identifier**. The **Ticks** represented by a value of CLR type "System.DateTime" MUST be the number of 100-nanosecond intervals that have elapsed since 12:00:00 midnight, January 1, 0001.

3.1.4.3.2.2 ReadEntityInstanceResponse

The **ReadEntityInstanceResponse** element specifies the result data for the **ReadEntityInstance** WSDL operation.

```
<xs:element name="ReadEntityInstanceResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="ReadEntityInstanceResult"
type="xs:boolean"/>
      <xs:element minOccurs="0" maxOccurs="1" name="ids" type="tns:ArrayOfString"/>
      <xs:element minOccurs="0" maxOccurs="1" name="displayName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="message" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="success" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

ReadEntityInstanceResult: MUST be set to **true** if an **EntityInstance** was identified based on the provided input.

ids: The list of serialized data values of the **EntityInstance** Identifiers corresponding to the retrieved **EntityInstance**. Each serialized data value of an **EntityInstance Identifier** MUST be encoded according to [\[XML10\]](#).

displayName: The display name corresponding to the retrieved **EntityInstance**. The value MUST be identified in the manner detailed in the description of the third element of the sub list contained in the values element **GetEntityInstancesResponse** (section [3.1.4.2.2.2](#)). The value MUST be encoded according to [\[XML10\]](#).

message: Contains errors or warnings regarding the process of reading the **EntityInstance** according to the provided input, if any. The message MUST be encoded as specified in [\[XML10\]](#).

success: A flag indicating whether the operation finished without any unexpected errors. The value MUST [<2>](#) be set to **true** if the operation finished without any unexpected errors; **false** otherwise. The value MUST be ignored by the protocol client. The value MUST be present.

3.1.4.3.3 Complex Types

None.

3.1.4.3.4 Simple Types

None.

3.1.4.3.5 Attributes

None.

3.1.4.3.6 Groups

None.

3.1.4.3.7 Attribute Groups

None.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4 Protocol Examples

This section provides specific example scenarios that illustrate the usage of the External Content Type Picker Web Service. The web service was designed to support the operations of the External Content Type Picker Control, which allows Microsoft Office users to pick or resolve EntityInstances. The picking scenario enables a user to perform a search to find and select an **EntityInstance** (GetEntityInstances) while the resolving scenario enables users to type a known key and resolve to get the desired **EntityInstance** (ReadEntityInstance).

The examples in this section use the following sample data. There is an Entity named "Customer" with fields (2) **CustomerID** and **CustomerName**. There are three **EntityInstances** available in the backend: Contoso with **CustomerID** 1 and **CustomerName** "Contoso", Fabrikam with **CustomerID** 2 and **CustomerName** "Fabrikam", Northwind with **CustomerID** 3 and **CustomerName** "Northwind".

Also, the Customer **Entity** has LobSystemInstance "ContosoCustomers" and **Entity** namespace <http://www.contoso.com>.

4.1 Getting Entity Instances

To get a set of EntityInstances for a given search token, protocol client constructs the following SOAP message:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <GetEntityInstances xmlns="http://tempuri.org/"
      xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
      <systemInstanceName>ContosoCustomers</systemInstanceName>
      <entityNamespace>http://www.contoso.com </entityNamespace>
      <entityName>Customer</entityName>
      <displayName>CustomerName</displayName>
      <searchToken/>
      <usedForPicking>true</usedForPicking>
      <maxResults>500</maxResults>
      <refreshInterval/>
    </GetEntityInstances>
  </s:Body>
</s:Envelope>
```

The protocol server then responds with the following:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"><GetEntityInstancesResponse
      xmlns="http://tempuri.org/">
    <GetEntityInstancesResult>3</GetEntityInstancesResult>
    <columnNames>
      <string>__identities</string>
      <string>__entityInstanceReference</string>
      <string>__displayName</string>
      <string>CustomerID</string>
      <string>CustomerName</string>
    </columnNames>
    <showInPicker>
      <boolean>>false</boolean>
      <boolean>>false</boolean>
      <boolean>>false</boolean>
      <boolean>>true</boolean>
      <boolean>>true</boolean>
    </showInPicker>
    <values>
```

```

    <string>__bg40001300</string>
    <string>22:http://www.contoso.com8:Customer16:CustomerRead
Item16:ContosoCustomersiAQAAAA==</string>
    <string>Contoso</string>
    <string>1</string>
    <string>Contoso</string>
    <string>__bg40002300</string>
    <string>22:http://www.contoso.com8:Customer16:CustomerRead
Item16:ContosoCustomersiAQBBB==</string>
    <string>Fabrikam</string>
    <string>2</string>
    <string>Fabrikam</string>
    <string>__bg40003300</string>
    <string>22:http://www.contoso.com8:Customer16:CustomerRead
Item16:ContosoCustomersiAQACCC==</string>
    <string>Northwind</string>
    <string>1</string>
    <string>Northwind</string>
  </values>
  <hasEntityMetadata>true</hasEntityMetadata>
  <message>Search criteria cannot be met as filters are not defined for the
Query. Contact your system administrator.</message>
  <success>true</success>
</GetEntityInstancesResponse>
</s:Body>
</s:Envelope>

```

4.2 Reading an Entity Instance

To read an EntityInstance, for example, when an instance needs to be resolved, the protocol client constructs the following SOAP message. In this example the reference to the **EntityInstance** is 22:http://www.contoso.com8:Customer16:CustomerRead Item16:ContosoCustomersiAQBBB==, so the protocol server returns the "Fabrikam" customer.

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <ReadEntityInstance xmlns="http://tempuri.org/"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
      <entityInstanceReference>22:http://www.contoso.com8:Customer16:CustomerRead
Item16:ContosoCustomersiAQBBB==</entityInstanceReference>
      <displayName>CustomerName</displayName>
      <fFormatAsXml>false</fFormatAsXml>
    </ReadEntityInstance></s:Body></s:Envelope>

```

The protocol server then responds with the Identifiers and display names:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"><s:Body
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <ReadEntityInstanceResponse xmlns="http://tempuri.org/">
    <ReadEntityInstanceResult>true</ReadEntityInstanceResult>
    <ids>
      <string>2</string>
    </ids>
    <displayName>Fabrikam</displayName>
    <success>true</success>
  </ReadEntityInstanceResponse></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:tns1="http://tempuri.org/Imports"
targetNamespace="http://tempuri.org/Imports">
      <xs:import namespace="http://tempuri.org/" />
    </xs:schema>
  </wsdl:types>
  <wsdl:portType name="IResolverPickerService">
    <wsdl:operation name="GetEntityInstances">
      <wsdl:input wsaw:Action="http://tempuri.org/IResolverPickerService/GetEntityInstances"
message="tns:IResolverPickerService_GetEntityInstances_InputMessage"/>
      <wsdl:output
wsaw:Action="http://tempuri.org/IResolverPickerService/GetEntityInstancesResponse"
message="tns:IResolverPickerService_GetEntityInstances_OutputMessage"/>
    </wsdl:operation>
    <wsdl:operation name="DecodeEntityInstanceId">
      <wsdl:input
wsaw:Action="http://tempuri.org/IResolverPickerService/DecodeEntityInstanceId"
message="tns:IResolverPickerService_DecodeEntityInstanceId_InputMessage"/>
      <wsdl:output
wsaw:Action="http://tempuri.org/IResolverPickerService/DecodeEntityInstanceIdResponse"
message="tns:IResolverPickerService_DecodeEntityInstanceId_OutputMessage"/>
    </wsdl:operation>
    <wsdl:operation name="ReadEntityInstance">
      <wsdl:input wsaw:Action="http://tempuri.org/IResolverPickerService/ReadEntityInstance"
message="tns:IResolverPickerService_ReadEntityInstance_InputMessage"/>
      <wsdl:output
wsaw:Action="http://tempuri.org/IResolverPickerService/ReadEntityInstanceResponse"
message="tns:IResolverPickerService_ReadEntityInstance_OutputMessage"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="DefaultBinding IResolverPickerService"
type="tns:IResolverPickerService">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="GetEntityInstances">
      <soap:operation
soapAction="http://tempuri.org/IResolverPickerService/GetEntityInstances" style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="DecodeEntityInstanceId">
      <soap:operation
soapAction="http://tempuri.org/IResolverPickerService/DecodeEntityInstanceId"
style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="ReadEntityInstance">
      <soap:operation
soapAction="http://tempuri.org/IResolverPickerService/ReadEntityInstance" style="document"/>
      <wsdl:input>
```

```

        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:message name="IResolverPickerService_DecodeEntityInstanceId_InputMessage">
    <wsdl:part name="parameters" element="tns:DecodeEntityInstanceId"/>
</wsdl:message>
<wsdl:message name="IResolverPickerService_DecodeEntityInstanceId_OutputMessage">
    <wsdl:part name="parameters" element="tns:DecodeEntityInstanceIdResponse"/>
</wsdl:message>
<wsdl:message name="IResolverPickerService_GetEntityInstances_InputMessage">
    <wsdl:part name="parameters" element="tns:GetEntityInstances"/>
</wsdl:message>
<wsdl:message name="IResolverPickerService_GetEntityInstances_OutputMessage">
    <wsdl:part name="parameters" element="tns:GetEntityInstancesResponse"/>
</wsdl:message>
<wsdl:message name="IResolverPickerService_ReadEntityInstance_InputMessage">
    <wsdl:part name="parameters" element="tns:ReadEntityInstance"/>
</wsdl:message>
<wsdl:message name="IResolverPickerService_ReadEntityInstance_OutputMessage">
    <wsdl:part name="parameters" element="tns:ReadEntityInstanceResponse"/>
</wsdl:message>
</wsdl:definitions>

```

7 Appendix B: Full XML Schema

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

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:tns="http://tempuri.org/" elementFormDefault="qualified"
targetNamespace="http://tempuri.org/" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="GetEntityInstances">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="siteId" type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="systemInstanceName" type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="entityNamespace" type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="entityName" type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="finderName" type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="displayFieldName" type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="searchToken" type="xs:string"/>
        <xs:element minOccurs="1" maxOccurs="1" name="usedForPicking" type="xs:boolean"/>
        <xs:element minOccurs="1" maxOccurs="1" name="maxResults" type="xs:unsignedInt"/>
        <xs:element minOccurs="1" maxOccurs="1" name="refreshInterval"
type="xs:unsignedInt"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="GetEntityInstancesResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="1" name="GetEntityInstancesResult"
type="xs:unsignedInt"/>
        <xs:element minOccurs="0" maxOccurs="1" name="columnNames" type="tns:ArrayOfString"/>
        <xs:element minOccurs="0" maxOccurs="1" name="localizedColumnNames"
type="tns:ArrayOfString"/>
        <xs:element minOccurs="0" maxOccurs="1" name="showInPicker"
type="tns:ArrayOfBoolean"/>
        <xs:element minOccurs="0" maxOccurs="1" name="values" type="tns:ArrayOfString"/>
        <xs:element minOccurs="1" maxOccurs="1" name="hasEntityMetadata" type="xs:boolean"/>
        <xs:element minOccurs="0" maxOccurs="1" name="message" type="xs:string"/>
        <xs:element minOccurs="1" maxOccurs="1" name="success" type="xs:boolean"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="ArrayOfString">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="string" nillable="true"
type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ArrayOfBoolean">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="boolean" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="DecodeEntityInstanceId">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="bstrSiteId" type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="bstrEntityInstanceId"
type="xs:string"/>
        <xs:element minOccurs="1" maxOccurs="1" name="fFormatAsXml" type="xs:boolean"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="DecodeEntityInstanceIdResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="DecodeEntityInstanceIdResult"
type="tns:ArrayOfString"/>
        <xs:element minOccurs="0" maxOccurs="1" name="message" type="xs:string"/>

```

```

        <xs:element minOccurs="1" maxOccurs="1" name="success" type="xs:boolean"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="ReadEntityInstance">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="siteId" type="xs:string"/>
            <xs:element minOccurs="0" maxOccurs="1" name="entityInstanceReference"
type="xs:string"/>
            <xs:element minOccurs="0" maxOccurs="1" name="displayFieldName" type="xs:string"/>
            <xs:element minOccurs="1" maxOccurs="1" name="fFormatAsXml" type="xs:boolean"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="ReadEntityInstanceResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="1" maxOccurs="1" name="ReadEntityInstanceResult"
type="xs:boolean"/>
            <xs:element minOccurs="0" maxOccurs="1" name="ids" type="tns:ArrayOfString"/>
            <xs:element minOccurs="0" maxOccurs="1" name="displayName" type="xs:string"/>
            <xs:element minOccurs="0" maxOccurs="1" name="message" type="xs:string"/>
            <xs:element minOccurs="1" maxOccurs="1" name="success" type="xs:boolean"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</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 Workspace 2010
- Microsoft Office 2010 suites
- Microsoft SharePoint Server 2010
- Microsoft Office 2013
- Microsoft SharePoint Server 2013
- Microsoft Office 2016
- 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.

[<1> Section 3.1.4.1.2.2](#): SharePoint Server 2010 always sets the value to false for this operation.

[<2> Section 3.1.4.3.2.2](#): SharePoint Server 2010 always sets the value to false for this operation.

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
8 Appendix C: Product Behavior	Updated list of supported products.	Y	Content updated due to protocol revision.

10 Index

A

Abstract data model
 [server](#) 13
[Applicability](#) 9
[ArrayOfString complex type](#) 12
[Attribute groups](#) 12
[Attributes](#) 12

C

[Capability negotiation](#) 9
[Change tracking](#) 39
Client
 [overview](#) 13
[Complex types](#) 11
 [ArrayOfString](#) 12

D

Data model - abstract
 [server](#) 13
[DecodeEntityInstanceId operation](#) 14

E

Events
 [local - server](#) 30
 [timer - server](#) 30
Example
 [overview](#) 31
Examples
 [getting Entity Instances](#) 31
 [reading an Entity Instance](#) 32

F

[Fields - vendor-extensible](#) 10
[Full WSDL](#) 34
[Full XML schema](#) 36

G

[GetEntityInstances operation](#) 19
[Getting Entity Instances example](#) 31
[Glossary](#) 6
[Groups](#) 12

I

[Implementer - security considerations](#) 33
[Index of security parameters](#) 33
[Informative references](#) 9
Initialization
 [server](#) 13
[Introduction](#) 6

L

Local events
 [server](#) 30

M

Message processing
 [server](#) 14
Messages
 [ArrayOfString complex type](#) 12
 [attribute groups](#) 12
 [attributes](#) 12
 [complex types](#) 11
 [elements](#) 11
 [enumerated](#) 11
 [groups](#) 12
 [namespaces](#) 11
 [simple types](#) 12
 [syntax](#) 11
 [transport](#) 11

N

[Namespaces](#) 11
[Normative references](#) 8

O

Operations
 [DecodeEntityInstanceId](#) 14
 [GetEntityInstances](#) 19
 [ReadEntityInstance](#) 27
[Overview \(synopsis\)](#) 9

P

[Parameters - security index](#) 33
[Preconditions](#) 9
[Prerequisites](#) 9
[Product behavior](#) 38
Protocol Details
 [overview](#) 13

R

[ReadEntityInstance operation](#) 27
[Reading an Entity Instance example](#) 32
[References](#) 8
 [informative](#) 9
 [normative](#) 8
[Relationship to other protocols](#) 9

S

Security
 [implementer considerations](#) 33
 [parameter index](#) 33
Sequencing rules
 [server](#) 14
Server
 [abstract data model](#) 13
 [DecodeEntityInstanceId operation](#) 14
 [GetEntityInstances operation](#) 19
 [initialization](#) 13

[local events](#) 30
[message processing](#) 14
[overview](#) 13
[ReadEntityInstance operation](#) 27
[sequencing rules](#) 14
[timer events](#) 30
[timers](#) 13
[Simple types](#) 12
[Standards assignments](#) 10
Syntax
[messages - overview](#) 11

T

Timer events
[server](#) 30
Timers
[server](#) 13
[Tracking changes](#) 39
[Transport](#) 11
Types
[complex](#) 11
[simple](#) 12

V

[Vendor-extensible fields](#) 10
[Versioning](#) 9

W

[WSDL](#) 34

X

[XML schema](#) 36