[MS-BDCDP]:
Business Data Catalog Data 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](https://www.microsoft.com/en-us/download/details.aspx?id=32985) or the [Microsoft Community Promise](https://www.microsoft.com/en-us/download/details.aspx?id=32985). 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.
- **License Programs.** To see all of the protocols in scope under a specific license program and the associated patents, visit the [Patent Map](https://www.microsoft.com/patents).
- **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](https://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.

Support. For questions and support, please contact dochelp@microsoft.com.
## Revision Summary

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision History</th>
<th>Revision Class</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>4/4/2008</td>
<td>0.1</td>
<td>New</td>
<td>Initial Availability</td>
</tr>
<tr>
<td>6/27/2008</td>
<td>1.0</td>
<td>Major</td>
<td>Revised and edited the technical content</td>
</tr>
<tr>
<td>12/12/2008</td>
<td>1.01</td>
<td>Editorial</td>
<td>Revised and edited the technical content</td>
</tr>
<tr>
<td>7/13/2009</td>
<td>1.02</td>
<td>Major</td>
<td>Changes made for template compliance</td>
</tr>
<tr>
<td>8/28/2009</td>
<td>1.03</td>
<td>Editorial</td>
<td>Revised and edited the technical content</td>
</tr>
<tr>
<td>11/6/2009</td>
<td>1.04</td>
<td>Editorial</td>
<td>Revised and edited the technical content</td>
</tr>
<tr>
<td>2/19/2010</td>
<td>2.0</td>
<td>Minor</td>
<td>Updated the technical content</td>
</tr>
<tr>
<td>3/31/2010</td>
<td>2.01</td>
<td>Editorial</td>
<td>Revised and edited the technical content</td>
</tr>
<tr>
<td>4/30/2010</td>
<td>2.02</td>
<td>Minor</td>
<td>Updated the technical content</td>
</tr>
<tr>
<td>6/7/2010</td>
<td>2.03</td>
<td>Editorial</td>
<td>Revised and edited the technical content</td>
</tr>
<tr>
<td>6/29/2010</td>
<td>2.04</td>
<td>Editorial</td>
<td>Changed language and formatting in the technical content.</td>
</tr>
<tr>
<td>7/23/2010</td>
<td>2.04</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>9/27/2010</td>
<td>2.04</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>11/15/2010</td>
<td>2.04</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>12/17/2010</td>
<td>2.04</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>3/18/2011</td>
<td>2.04</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>6/10/2011</td>
<td>2.04</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>1/20/2012</td>
<td>3.0</td>
<td>Major</td>
<td>Significantly changed the technical content</td>
</tr>
<tr>
<td>4/11/2012</td>
<td>3.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>7/16/2012</td>
<td>3.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>9/12/2012</td>
<td>3.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>10/8/2012</td>
<td>3.1</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content</td>
</tr>
<tr>
<td>2/11/2013</td>
<td>3.1</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>7/30/2013</td>
<td>3.2</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content</td>
</tr>
<tr>
<td>11/18/2013</td>
<td>3.2</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>Date</td>
<td>Revision History</td>
<td>Revision Class</td>
<td>Comments</td>
</tr>
<tr>
<td>------------</td>
<td>------------------</td>
<td>----------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>2/10/2014</td>
<td>3.2</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>4/30/2014</td>
<td>3.2</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>7/31/2014</td>
<td>3.2</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>10/30/2014</td>
<td>3.2</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>2/26/2016</td>
<td>4.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>7/15/2016</td>
<td>4.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>9/14/2016</td>
<td>4.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>10/17/2016</td>
<td>4.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>7/24/2018</td>
<td>5.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>10/1/2018</td>
<td>6.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>3/19/2019</td>
<td>6.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>6/18/2019</td>
<td>6.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
</tbody>
</table>
Table of Contents

1 Introduction ................................................................................. 6
  1.1 Glossary ................................................................................. 6
  1.2 References ............................................................................. 7
    1.2.1 Normative References ...................................................... 7
    1.2.2 Informative References .................................................... 8
  1.3 Overview ............................................................................... 8
  1.4 Relationship to Other Protocols .............................................. 8
  1.5 Prerequisites/Preconditions .................................................... 9
  1.6 Applicability Statement ........................................................ 9
  1.7 Versioning and Capability Negotiation .................................... 9
  1.8 Vendor-Extensible Fields ....................................................... 9
  1.9 Standards Assignments ........................................................ 9

2 Messages .................................................................................. 10
  2.1 Transport .............................................................................. 10
  2.2 Common Message Syntax ..................................................... 10
    2.2.1 Namespaces .................................................................... 10
    2.2.2 Messages ....................................................................... 10
    2.2.3 Elements ....................................................................... 11
    2.2.4 Complex Types ................................................................ 11
    2.2.5 Simple Types ................................................................... 11
    2.2.6 Attributes ....................................................................... 11
    2.2.7 Groups ......................................................................... 11
    2.2.8 Attribute Groups ............................................................ 11

3 Protocol Details .......................................................................... 12
  3.1 Protocol Server Details ........................................................ 12
    3.1.1 Abstract Data Model ......................................................... 12
    3.1.2 Timers ............................................................................ 12
    3.1.3 Initialization .................................................................... 12
    3.1.4 Message Processing Events and Sequencing Rules ............. 12
      3.1.4.1 Resolve ................................................................. 13
        3.1.4.1.1 Messages .......................................................... 13
          3.1.4.1.1.1 ResolveSoapIn .............................................. 13
          3.1.4.1.1.2 ResolveSoapOut .......................................... 14
        3.1.4.1.2 Elements ........................................................... 14
          3.1.4.1.2.1 Resolve ......................................................... 14
          3.1.4.1.2.2 ResolveResponse ......................................... 14
        3.1.4.1.3 Complex Types .................................................. 15
          3.1.4.1.3.1 ResolveResult .............................................. 15
          3.1.4.1.3.2 IdentifierField ............................................. 15
          3.1.4.1.3.3 ArrayOfFieldRecord ................................... 16
          3.1.4.1.3.4 FieldRecord ............................................... 16
        3.1.4.1.4 Simple Types ..................................................... 16
          3.1.4.1.4.1 ResolveStatus .............................................. 16
        3.1.4.1.5 Attributes ........................................................... 17
        3.1.4.1.6 Groups ............................................................... 17
        3.1.4.1.7 Attribute Groups ................................................. 17
    3.1.5 Timer Events ...................................................................... 17
    3.1.6 Other Local Events ........................................................ 17

4 Protocol Examples ........................................................................ 18
  4.1 Retrieving Field Values for LobSystem Entities ....................... 18

5 Security ..................................................................................... 19
  5.1 Security Considerations for Implementers ................................. 19
5.2  Index of Security Parameters ................................................................. 19
6  Appendix A: Full WSDL ........................................................................... 20
7  Appendix B: Product Behavior ................................................................. 22
8  Change Tracking ......................................................................................... 23
9  Index ............................................................................................................. 24
1 Introduction

The Business Data Catalog Data Web Service Protocol is an interface that protocol clients can use to search software systems that store business data and process rules for an instance of a particular entity.

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:

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

- **EntityInstanceId**: A set of Field values of an EntityInstance that collectively and uniquely identify an EntityInstance in a line-of-business (LOB) system.

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

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

- **line-of-business (LOB) system**: A software system that is used to store business data and can also contain business rules and business logic that support business processes.

- **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 that represent a LobSystem, LobSystemInstance, DataClass, Entity, Method, MethodInstance, Parameter, TypeDescriptor, Identifier, FilterDescriptor, Action, ActionParameter, or Association.

- **Model**: A MetadataObject that defines a set of references to a logically related set of Entities. Models are typically used to facilitate easy transfer into and out of Metadata stores.

- **site**: A group of related pages and data within a SharePoint site collection. The structure and content of a site is based on a site definition. Also referred to as SharePoint site and web site.

- **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.

**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 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](mailto:dochelp@microsoft.com). We will assist you in finding the relevant information.


1.3 Overview

This document specifies a consistent selection process for information stored in any **line-of-business (LOB) system** for protocol client retrieval.

Enterprises store a variety of information in various line-of-business (LOB) systems. This protocol specifies unified information selection for all types of **LobSystems**. The protocol enables consistent processing for the user experience because it describes **LobSystems** in terms of **Models**, **LobSystemInstances**, and the business data types they store as **Entities**. It also describes how the protocol client selects information from the protocol server.

This protocol enables a protocol client to retrieve an **EntityInstanceId** and the values for a list of **field** names for an **Entity** associated with a **LobSystemInstance**. The protocol allows the protocol client to send the **LobSystemInstance** name, the **Entity** name, a value to resolve, and a list of **Entity** fields to the protocol server, and to receive the serialized **EntityInstanceId** and the list of **Entity** field name-value pairs.

1.4 Relationship to Other Protocols

This protocol uses the SOAP messaging protocol for formatting requests and responses as described either in **[SOAP1.1]** or in **[SOAP1.2-1/2007]** and **[SOAP1.2-2/2007]**. It transmits these messages using the HTTP protocol as described in **[RFC2616]** or the HTTPS protocol as described in **[RFC2818]**.

The following diagram shows the underlying messaging and transport stack that the protocol uses:
1.5 Prerequisites/Preconditions

This protocol operates against a site that is known to protocol clients by its URL. The protocol client forms an endpoint by appending "/_vti_bin/bdcfieldsresolver.asmx" to the URL of the site, for example "http://www.contoso.com/Repository/_vti_bin/bdcfieldsresolver.asmx". This protocol assumes that the underlying protocols perform authentication.

1.6 Applicability Statement

None.

1.7 Versioning and Capability Negotiation

This document describes versioning issues in the following areas:

- **Supported transports:** 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 additionally support SOAP over HTTPS to secure communication with protocol clients.

Protocol messages MUST be formatted as specified either in [SOAP1.1], section 4 or in [SOAP1.2-1/2007] section 5. Protocol server faults MUST be returned using HTTP Status Codes as specified in [RFC2616] section 10, or using SOAP faults as specified in [SOAP1.1] section 4.4 or in [SOAP1.2-1/2007] section 5.4.

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 [XMLSCHEMA1/2] and [XMLSCHEMA2/2], and WSDL, as specified in [WSDL].

2.2.1 Namespaces

This protocol specifies and references XML namespaces using the mechanisms specified in [XMLNS]. Although this document associates an 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.

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Namespace URI</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>soap</td>
<td><a href="http://schemas.xmlsoap.org/wsd1/soap/">http://schemas.xmlsoap.org/wsd1/soap/</a></td>
<td>[SOAP1.1]</td>
</tr>
<tr>
<td>s</td>
<td><a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a></td>
<td>[XMLSCHEMA1/2] [XMLSCHEMA2/2]</td>
</tr>
<tr>
<td>soap12</td>
<td><a href="http://schemas.xmlsoap.org/wsd1/soap12/">http://schemas.xmlsoap.org/wsd1/soap12/</a></td>
<td>[SOAP1.2-1/2007] [SOAP1.2-2/2007]</td>
</tr>
<tr>
<td>(none)</td>
<td><a href="http://microsoft.com/webservices/SharePointPortalServer/BDCClientWS/">http://microsoft.com/webservices/SharePointPortalServer/BDCClientWS/</a></td>
<td></td>
</tr>
<tr>
<td>wsd1</td>
<td><a href="http://schemas.xmlsoap.org/wsd1/">http://schemas.xmlsoap.org/wsd1/</a></td>
<td>[WSDL]</td>
</tr>
<tr>
<td>mime</td>
<td><a href="http://schemas.xmlsoap.org/wsd1/mime/">http://schemas.xmlsoap.org/wsd1/mime/</a></td>
<td>[WSDL]</td>
</tr>
<tr>
<td>http</td>
<td><a href="http://schemas.xmlsoap.org/wsd1/http/">http://schemas.xmlsoap.org/wsd1/http/</a></td>
<td>[WSDL]</td>
</tr>
<tr>
<td>tm</td>
<td><a href="http://microsoft.com/wsd1/mime/textMatching/">http://microsoft.com/wsd1/mime/textMatching/</a></td>
<td>[WSDL]</td>
</tr>
<tr>
<td>soapenc</td>
<td><a href="http://schemas.xmlsoap.org/soap/encoding/">http://schemas.xmlsoap.org/soap/encoding/</a></td>
<td></td>
</tr>
</tbody>
</table>

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

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

This protocol operates between the protocol client and a protocol server. The protocol client initiates the communication, and the protocol server responds. The protocol server does not retain any states.

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.

Except where specified, protocol clients interpret HTTP status codes returned by the protocol server as specified in [RFC2616] section 10.

Protocol servers perform implementation-specific authorization checks and notify protocol clients of authorization faults using either HTTP status codes or SOAP faults.

3.1 Protocol 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 the following MetadataObject types: LobSystem, LobSystemInstance, and Entity. The protocol server maintains the following relationships between these MetadataObject types:

- LobSystems contain all Entities.
- LobSystems contain all LobSystemInstances.

The protocol server MUST assign unique names to the following MetadataObjects.

- All Entities contained by a particular LobSystem.
- All LobSystemInstances contained by a metadata store.
- All LobSystems contained by a metadata store.

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 WSDL operations as defined by this specification:

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
</table>

[MS-BDCDP] - v20190618
Business Data Catalog Data Web Service Protocol
Copyright © 2019 Microsoft Corporation
Release: June 18, 2019
### Operation Description

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolve</td>
<td>This operation retrieves the values of a list of Entity fields (4) associated with the EntityInstance that was specified in the input criteria.</td>
</tr>
</tbody>
</table>

#### 3.1.4.1 Resolve

This operation searches for an EntityInstance that matches the specified input criteria. The input criteria are the LobSystemInstance name, the Entity name and value to resolve. The mechanism used by the protocol server to match the EntityInstances is implementation-specific.

```xml
<wsdl:operation name="Resolve">
  <wsdl:input message="tns:ResolveSoapIn" />
  <wsdl:output message="tns:ResolveSoapOut" /> 
</wsdl:operation>
```

The protocol client sends a ResolveSoapIn request message and the protocol server responds with a ResolveSoapOut response message. The protocol client MUST specify the following:

- The name of the LobSystemInstance to search.
- The name of the Entity to search.
- The search string that represents the value to resolve.
- The field names of the Entity for which to retrieve the values of the specified EntityInstance.

If exactly one EntityInstance is found, the protocol server sends the serialized EntityInstanceId of the specified EntityInstance, along with the list of name-value pairs for the fields of the specified Entity and the status of the operation.

This operation MUST return a ResolveResult message with the ResolveStatus element set, as specified in section 3.1.4.1.3.1.

#### 3.1.4.1.1 Messages

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResolveSoapIn</td>
<td>A request message for the Resolve operation.</td>
</tr>
<tr>
<td>ResolveSoapOut</td>
<td>The response message for the Resolve operation.</td>
</tr>
</tbody>
</table>

#### 3.1.4.1.1.1 ResolveSoapIn

The ResolveSoapIn message is the request message for the Resolve operation.

The SOAP action value of the message is:

```xml
```
The SOAP body contains a Resolve element.

### 3.1.4.1.2 ResolveSoapOut

The ResolveSoapOut message is the response message for the Resolve operation.

The SOAP body contains a ResolveResponse element that MUST specify the status of the operation.

### 3.1.4.1.2 Elements

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolve</td>
<td>A message containing the input data for the Resolve operation.</td>
</tr>
<tr>
<td>ResolveResponse</td>
<td>A message containing the output data of a Resolve operation.</td>
</tr>
</tbody>
</table>

#### 3.1.4.1.2.1 Resolve

The Resolve message contains the input data for the Resolve operation.

```
<s:element name="Resolve">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="systemInstance" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="entity" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="valueToResolve" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="fieldNames" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
```

**systemInstance:** This element contains the name of the LobSystemInstance. The value is encoded as specified in [XML10]. The protocol server MUST only match EntityInstances obtained from the specified LobSystemInstance.

**entity:** This element contains the name of the Entity. The value is encoded as specified in [XML10]. The protocol server MUST only match EntityInstances of the specified Entity.

**valueToResolve:** This element contains the value to resolve to an EntityInstance. The value is encoded as specified in [XML10]. The value is resolved to EntityInstances using an implementation-specific algorithm. For example, a specific implementation of a protocol server could resolve a given value of "ab" and Entity "Customer", by returning customers that have names starting with "ab". Another implementation of the protocol server could return customer whose names, last names, or addresses contain "ab".

**fieldNames:** This element contains the field names associated with the Entity. Each field name is encoded as specified in [RFC3986]. The field names are concatenated into a single string with a colon (:) between field names.

#### 3.1.4.1.2.2 ResolveResponse

The ResolveResponse message contains the output data of a Resolve operation.
<s:element name="ResolveResponse">
  <s:complexType>
  <s:sequence>
  <s:element minOccurs="1" maxOccurs="1" name="ResolveResult" type="tns:ResolveResult" />
  </s:sequence>
  </s:complexType>
</s:element>

**ResolveResult**: This element contains the serialized **EntityInstanceId** element that is associated with the specified **EntityInstance** element, along with the list of name-value pairs for the **fields** of the specified **Entity** and the status of the operation.

### 3.1.4.1.3 Complex Types

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

<table>
<thead>
<tr>
<th>Complex type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResolveResult</td>
<td>A message containing the output data of a <strong>Resolve</strong> operation.</td>
</tr>
<tr>
<td>IdentifierField</td>
<td>An element containing the <strong>Identifier</strong> of the <strong>EntityInstance</strong>.</td>
</tr>
<tr>
<td>ArrayOfFieldRecord</td>
<td>An element containing an array of elements of type <strong>FieldRecord</strong>.</td>
</tr>
<tr>
<td>FieldRecord</td>
<td>An element containing the <strong>Entity</strong> field (4) name-value pair.</td>
</tr>
</tbody>
</table>

#### 3.1.4.1.3.1 ResolveResult

The **ResolveResult** message contains the output data of a **Resolve** operation.

```xml
<s:complexType name="ResolveResult">
  <s:sequence>
  <s:element minOccurs="0" maxOccurs="1" name="Identifier" type="tns:IdentifierField" />
  <s:element minOccurs="0" maxOccurs="1" name="Results" type="tns:ArrayOfFieldRecord" />
  <s:element minOccurs="1" maxOccurs="1" name="Status" type="tns:ResolveStatus" />
  </s:sequence>
</s:complexType>
```

**Identifier**: This element contains the **Identifier** of the **EntityInstance** found. If exactly one **EntityInstance** is not found, the value MUST be absent.

**Results**: This element contains the list of name-value pairs of the **fields** of the specified **Entity** if the **fieldNames** element is not absent. If exactly one **EntityInstance** element is not found, the value MUST be absent.

**Status**: This element contains the status of the operation. This **ResolveStatus** element MUST be present. The protocol server MUST set this value as specified in section 3.1.4.1.4.1.

#### 3.1.4.1.3.2 IdentifierField

The **IdentifierField** element contains the **Identifier** of the **EntityInstance**. If exactly one **EntityInstance** is not found, the value MUST be absent.
3.1.4.1.3.3 ArrayOfFieldRecord

This `ArrayOfFieldRecord` element contains an array of elements of type `FieldRecord`, as specified in section 3.1.4.1.3.4.

```xml
<s:complexType name="ArrayOfFieldRecord">
  <s:sequence>
    <s:element name="FieldRecord" type="tns:FieldRecord" nillable="true" minOccurs="0" maxOccurs="unbounded"/>
  </s:sequence>
</s:complexType>
```

**FieldRecord**: This element contains the **Entity field** name-value pair.

3.1.4.1.3.4 FieldRecord

The `FieldRecord` element contains the **Entity field** name-value pair. The name of the field is contained in the `FieldName` attribute of the element and the value of the field is the value of the element.

```xml
<s:complexType name="FieldRecord">
  <s:simpleContent>
    <s:extension base="s:string">
      <s:attribute name="FieldName" type="s:string"/>
    </s:extension>
  </s:simpleContent>
</s:complexType>
```

**FieldName**: This attribute contains the name of the field. Each field name is encoded as specified in [RFC3986].

3.1.4.1.4 Simple Types

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

<table>
<thead>
<tr>
<th>Simple type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResolveStatus</td>
<td>An enumeration that specifies the status of the Resolve operation.</td>
</tr>
</tbody>
</table>

3.1.4.1.4.1 ResolveStatus

The `ResolveStatus` simple type specifies the enumeration of status of the Resolve operation, as specified in section 3.1.4.1.

```xml
<s:simpleType name="ResolveStatus">
  <s:restriction base="s:string">
    <s:enumeration value="NoMatch"/>
  </s:restriction>
</s:simpleType>
```
The values for this enumeration are specified in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NoMatch</td>
<td>The protocol server MUST return this value when there are no EntityInstances were found.</td>
</tr>
<tr>
<td>MultipleMatch</td>
<td>The protocol server MUST return this value when more than one EntityInstances were found.</td>
</tr>
<tr>
<td>UniqueMatch</td>
<td>The protocol server MUST return this value when exactly one EntityInstance was found.</td>
</tr>
<tr>
<td>InvalidData</td>
<td>The protocol server MUST return this value when the operation has failed because input data is not valid or an implementation-specific error has occurred.</td>
</tr>
</tbody>
</table>

3.1.4.1.5 Attributes

None.

3.1.4.1.6 Groups

None.

3.1.4.1.7 Attribute Groups

None.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.
4 Protocol Examples

4.1 Retrieving Field Values for LobSystem Entities

This scenario resolves the specified value to an **EntityInstance**. For this example, the protocol client is aware of the **LobSystemInstance** name, the Entity name, the value to resolve and the **Entity field** names. The steps are as follows.

1. The protocol client encodes and concatenates the **Entity** field names using the colon (:) as a separator.
   
   If the values of the fields **ProductKey, ProductName, Price** and **Color** are to be obtained, the client prepares a string of the form "ProductKey:ProductName:Price:Color"

2. The protocol client issues a **Resolve** request to the protocol server with the **LobSystemInstance** name, the **Entity** name, the value to resolve and the concatenated **Entity** field names.

   ```xml
   <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema"><soap:Body><Resolve
   xmlns="http://microsoft.com/webservices/SharePointPortalServer/BDCClientWS/"
   instance>bdcdpExampleInstance</systemInstance><entity>Product</entity><valueToResolve>
1</valueToResolve><fieldNames>ProductKey:ProductName:Price:Color</fieldNames></Resolve>
   </soap:Body></soap:Envelope>
   
   3. The protocol server responds with the status of the operation. If the **EntityInstance** is found, the protocol server also returns the specified **EntityInstance** and the name-value pairs for the fields.

   ```xml
   "<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema"><ResolveResponse
   xmlns="http://microsoft.com/webservices/SharePointPortalServer/BDCClientWS/"<Resolve
   Result><Identifier>_bg40001300</Identifier><Results><FieldRecord
   FieldName="ProductKey">1</FieldRecord><FieldRecord
   FieldName="ProductName">Bag</FieldRecord><FieldRecord
   FieldName="Price">3.0000</FieldRecord><FieldRecord
   FieldName="Color">Yellow</FieldRecord></Results><Status>UniqueMatch</Status></Resolve
   Result></ResolveResponse></soap:Body"></soap:Envelope>
   ```
5 Security

5.1 Security Considerations for Implementers
None.

5.2 Index of Security Parameters
None.
Appendix A: Full WSDL

For ease of implementation, the full WSDL and schema are provided in this appendix.

```xml
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
  <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    BDC Field Resolver Web Service for Clients
  </wsdl:documentation>
  <wsdl:types>
    <s:schema targetNamespace="http://microsoft.com/webservices/SharePointPortalServer/BDCClientWS/">
      <s:element name="Resolve">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="systemInstance" type="s:string"/>
            <s:element minOccurs="1" maxOccurs="1" name="entity" type="s:string"/>
            <s:element minOccurs="1" maxOccurs="1" name="valueToResolve" type="s:string"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="ResolveResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="ResolveResult" type="tns:ResolveResult"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="ResolveResult">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="Identifier" type="tns:IdentifierField"/>
          <s:element minOccurs="0" maxOccurs="1" name="Results" type="tns:ArrayOfFieldRecord"/>
        </s:sequence>
      </s:complexType>
      <s:complexType name="IdentifierField">
        <s:simpleContent>
          <s:extension base="s:string">
            <s:attribute name="FieldName" type="s:string"/>
          </s:simpleContent>
        </s:complexType>
      </s:complexType>
      <s:complexType name="ArrayOfFieldRecord">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="unbounded" name="FieldRecord" nillable="true" type="tns:FieldRecord"/>
        </s:sequence>
      </s:complexType>
      <s:complexType name="FieldRecord">
        <s:simpleContent>
          <s:extension base="s:string"/>
        </s:complexType>
      </s:complexType>
    </s:schema>
  </wsdl:types>
</wsdl:definitions>
```
<s:simpleType name="ResolveStatus">
  <s:restriction base="s:string">
    <s:enumeration value="NoMatch" />
    <s:enumeration value="MultipleMatch" />
    <s:enumeration value="UniqueMatch" />
    <s:enumeration value="InvalidData" />
  </s:restriction>
</s:simpleType>

<wSDL:types>
  <wSDL:message name="ResolveSoapIn">
    <wSDL:part name="parameters" element="tns:Resolve" />
  </wSDL:message>
  <wSDL:message name="ResolveSoapOut">
    <wSDL:part name="parameters" element="tns:ResolveResponse" />
  </wSDL:message>
  <wSDL:portType name="BDCFieldsResolverSoap">
    <wSDL:operation name="Resolve">
      <wSDL:input message="tns:ResolveSoapIn" />
      <wSDL:output message="tns:ResolveSoapOut" />
    </wSDL:operation>
  </wSDL:portType>
  <wSDL:binding name="BDCFieldsResolverSoap" type="tns:BDCFieldsResolverSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wSDL:operation name="Resolve">
      <soap:operation
      <wSDL:input>
        <soap:body use="literal" />
      </wSDL:input>
      <wSDL:output>
        <soap:body use="literal" />
      </wSDL:output>
    </wSDL:operation>
  </wSDL:binding>
  <wSDL:binding name="BDCFieldsResolverSoap12" type="tns:BDCFieldsResolverSoap">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wSDL:operation name="Resolve">
      <soap12:operation
      <wSDL:input>
        <soap12:body use="literal" />
      </wSDL:input>
      <wSDL:output>
        <soap12:body use="literal" />
      </wSDL:output>
    </wSDL:operation>
  </wSDL:binding>
</wSDL:types>
7 Appendix B: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include updates to those products.

- Microsoft Office SharePoint Server 2007
- Microsoft SharePoint Server 2010
- Microsoft SharePoint Server 2013
- Microsoft SharePoint Server 2016
- Microsoft SharePoint Server 2019

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates 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.
8 Change Tracking

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

A
Abstract data model
server 12
Applicability 9
Attribute groups 11
Attributes 11

C
Capability negotiation 9
Change tracking 23
Client
overview 12
Complex types 11

D
Data model - abstract
server 12

E
Events
local - server 17
timer - server 17
Examples
retrieving field values for LobSystem Entities 18

F
Fields - vendor-extensible 9
Full WSDL 20

G
Glossary 6
Groups 11

I
Implementer - security considerations 19
Index of security parameters 19
Informative references 8
Initialization
server 12
Introduction 6

L
Local events
server 17

M
Message processing
server 12
Messages
attribute groups 11
attributes 11
complex types 11

elements 11
enumerated 11
groups 11
namespaces 10
simple types 11
syntax 10
transport 10

N
Namespaces 10
Normative references 7

O
Operations
Resolve 13
Overview (synopsis) 8

P
Parameters - security index 19
Preconditions 9
Prerequisites 9
Product behavior 22
Protocol Details
overview 12

R
References 7
informative 8
normative 7
Relationship to other protocols 8
Retrieving field values for LobSystem Entities
example 18

S
Security
implementer considerations 19
parameter index 19
Sequencing rules
server 12
Server
abstract data model 12
initialization 12
local events 17
message processing
overview 12
Resolve operation 13
sequencing rules 12
timer events 17
timers 12
Simple types 11
Standards assignments 9
Syntax
messages - overview 10

T
Timer events