[MS-ASDT]:

Access Server Design Time Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
1/20/2012	0.1	New	Released new document.
4/11/2012	0.1	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	0.1	None	No changes to the meaning, language, or formatting of the technical content.
9/12/2012	0.1	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	1.0	Major	Significantly changed the technical content.
2/11/2013	1.0	None	No changes to the meaning, language, or formatting of the technical content.
7/30/2013	1.0	None	No changes to the meaning, language, or formatting of the technical content.
11/18/2013	1.0	None	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	1.0	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	1.0	None	No changes to the meaning, language, or formatting of the technical content.
7/31/2014	1.0	None	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	1.0	None	No changes to the meaning, language, or formatting of the technical content.
2/26/2016	2.0	Major	Significantly changed the technical content.
7/15/2016	2.0	None	No changes to the meaning, language, or formatting of the technical content.
9/14/2016	2.0	None	No changes to the meaning, language, or formatting of the technical content.
7/24/2018	3.0	Major	Significantly changed the technical content.
10/1/2018	4.0	Major	Significantly changed the technical content.
7/20/2021	5.0	Major	Significantly changed the technical content.
8/17/2021	6.0	Major	Significantly changed the technical content.
2/15/2022	6.0	None	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction

The Access Server Design Time Protocol enables a protocol client to create **database applications** that are stored on the 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:

collation: A set of rules that determines how data is compared, ordered, and presented.

culture name: A part of a language identification tagging system, as described in [RFC1766]. Culture names adhere to the format "<languagecode2>-<country/regioncode2>." If a two-letter language code is not available, a three-letter code that is derived from [ISO-639] is used.

database application: A set of objects, including tables, queries, forms, reports, macros, and code modules, that are stored in a database structure.

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

fully qualified URL: A URL that includes a protocol scheme name, a host name, optionally a port number, a path, optionally a search part, and optionally a fragment identifier, as described in [RFC2616].

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].

IPv4 address in string format: A string representation of an IPv4 address in dotted-decimal notation, as described in [RFC1123] section 2.1.

IPv6 address in string format: A string representation of an IPv6 address, as described in [RFC4291] section 2.2.

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: 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.

SOAP fault code: The algorithmic mechanism for identifying a **SOAP fault**. See [SOAP1.2-1/2007] section 5.6 for more information.

surrogate pair: A pair of 16-bit Unicode encoding values that, together, represent a single 32-bit character, as described in [ISO-10646]. For more information about surrogate pairs and combining character sequences, see the Unicode Standard in [UNICODE].

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

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

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

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

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

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

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

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

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.

[MS-AADT] Microsoft Corporation, "Access Application Design Time Protocol".

[RFC1123] Braden, R., "Requirements for Internet Hosts - Application and Support", RFC 1123, October 1989, http://www.ietf.org/rfc/rfc1123.txt

[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

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000, http://www.rfc-editor.org/rfc/rfc2818.txt

[RFC4291] Hinden, R. and Deering, S., "IP Version 6 Addressing Architecture", RFC 4291, February 2006, http://www.ietf.org/rfc/rfc4291.txt

[RFC4646] Phillips, A., and Davis, M., Eds., "Tags for Identifying Languages", BCP 47, RFC 4646, September 2006, http://www.rfc-editor.org/rfc/rfc4646.txt

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

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

[SOAP1.2-2/2007] Gudgin, M., Hadley, M., Mendelsohn, N., et al., "SOAP Version 1.2 Part 2: Adjuncts (Second Edition)", W3C Recommendation, April 2007, http://www.w3.org/TR/2007/REC-soap12-part2-20070427

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

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

[XMLSCHEMA1/2] Thompson, H., Beech, D., Maloney, M., and Mendelsohn, N., Eds., "XML Schema Part 1: Structures Second Edition", W3C Recommendation, October 2004, http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/

[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/

[XMLSCHEMA2/2] Biron, P., and Malhotra, A., Eds., "XML Schema Part 2: Datatypes Second Edition", W3C Recommendation, October 2004, http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/

[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 create **database applications** on a protocol server. The database applications created using this protocol allow a user to enter, store, view, and analyze data through a Web browser. Additionally, this protocol enables a protocol client to obtain versioning information about the protocol server for use when designing database applications.

1.4 Relationship to Other Protocols

The Access Server Design Time Protocol uses **SOAP** over **HTTP**, as described in [RFC2616], and SOAP over **HTTPS**, as described in [RFC2818], as shown in the following layering diagram:

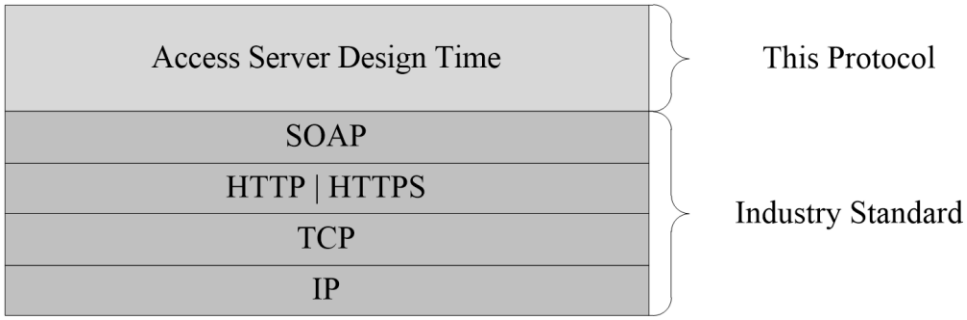

Figure 1: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

This protocol operates against a **site** that is identified by a **URI** that is known by protocol clients. The protocol server **endpoint** is formed by appending "/_vti_bin/accsvc/ServerDesignService.asmx" to the URI of the site, for example: http://www.example.com/Repository/_vti_bin/accsvc/ServerDesignService.asmx.

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

1.6 Applicability Statement

This protocol is applicable in the following scenarios:

- Creation of **database applications** that will be designed by using the Access Application Design Time Protocol, as described in [MS-AADT].
- Querying a protocol server for the **Access Services Protocol Versions** (section 3.1.1.1) it supports, which will then be used by various messages in both this protocol and the Access Application Design Time Protocol, as described in [MS-AADT].

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- **Supported Transports:** This protocol can be implemented by using transports that support sending **SOAP** version 1.2 messages, as specified in section 2.1.

- **Protocol Versions:** The only message this protocol versions is the **CreateApplication** message specified in section 3.1.4.1, which is specified by an **Access Services Protocol Version** (section 3.1.1.1). All other messages are not versioned.
- **Capability Negotiation:** When calling **CreateApplication**, the protocol client sends a list of **Access Services Protocol Versions** it supports, as specified in section 3.1.4.1.3.1. The protocol server chooses one and includes its choice with its response to the protocol client, as specified in section 3.1.4.1.3.3. Alternatively, a protocol client retrieves the **Access Services Protocol Versions** the protocol server supports by calling **GetServerInformation**, as specified in section 3.1.4.2.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

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

2.1 Transport

Messages MUST be transported by using **SOAP** version 1.2, as specified in [SOAP1.2-1/2007] and [SOAP1.2-2/2007], over **HTTP**, as specified in [RFC2616], or **HTTPS**, as specified in [RFC2818].

2.2 Common Message Syntax

This section contains common definitions used by this protocol. The syntax of the definitions uses **XML schema** as defined in [XMLSCHEMA1] and [XMLSCHEMA2], and **WSDL** as defined 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
http	http://schemas.xmlsoap.org/wsdl/http/	
soap	http://schemas.xmlsoap.org/wsdl/soap/	[SOAP1.1]
soap12	http://schemas.xmlsoap.org/wsdl/soap12/	[SOAP1.2-1/2007] [SOAP1.2-2/2007]
tns	http://schemas.microsoft.com/office/Access/2010/11/Server/WebServices/ServerDesignService	
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1/2] [XMLSCHEMA2/2]

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
ProtocolVersionList	A sequence of **Access Services Protocol Versions**, as specified by section 3.1.1.1.
ServiceResult	The base complex type extended by all response messages defined in this protocol.
ServiceWarning	Specifies additional information as part of a successful response from the protocol server.

2.2.4.1 ProtocolVersionList

Namespace:
http://schemas.microsoft.com/office/Access/2010/11/Server/WebServices/ServerDesignService

A sequence of **Access Services Protocol Versions**, as specified by section 3.1.1.1.

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

ProtocolVersion: A **string** ([XMLSCHEMA2] section 3.2.1) that specifies an **Access Services Protocol Version** (section 3.1.1.1). MUST be present.

2.2.4.2 ServiceResult

Namespace:
http://schemas.microsoft.com/office/Access/2010/11/Server/WebServices/ServerDesignService

The base complex type extended by all response messages defined in this protocol.

```
<xs:complexType name="ServiceResult" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="1" name="Warning" nillable="true"
type="tns:ServiceWarning"/>
  </xs:sequence>
</xs:complexType>
```

Warning: A **ServiceWarning** (section 2.2.4.3) that specifies additional information that accompanies the response. MUST be nil as specified by [XMLSCHEMA1] section 2.6.2 in all responses returned by this protocol.

2.2.4.3 ServiceWarning

Namespace:
http://schemas.microsoft.com/office/Access/2010/11/Server/WebServices/ServerDesignService

Specifies additional information as part of a successful response from the protocol server.

```
<xs:complexType name="ServiceWarning" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:attribute name="Code" type="xs:string"/>
  <xs:attribute name="Message" type="xs:string"/>
</xs:complexType>
```

Code: A **string** ([XMLSCHEMA2] section 3.2.1) that specifies an identifier of the warning, similar to a **SOAP fault code**.

Message: A **string** ([XMLSCHEMA2] section 3.2.1) that specifies a message that corresponds to the **Code**. The value of this message is implementation-dependent.

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. 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.

3.1.1.1 Access Services Protocol Version

The **Access Services Protocol Version** is a token from the protocol server that the protocol client and server use to determine whether they are using compatible versions of the protocol. MUST be a **string** ([XMLSCHEMA2] section 3.2.1). MUST be one of the following values:

Value	Meaning
15.0.21.0	The protocol client and server are using the following **XML namespaces**: http://schemas.microsoft.com/office/accessservices/2010/12/application
15.0.22.0	The protocol client and server are using the following XML namespaces: http://schemas.microsoft.com/office/accessservices/2010/12/application
15.0.23.0	The protocol client and server are using the following XML namespaces: http://schemas.microsoft.com/office/accessservices/2010/12/application
15.0.24.0	The protocol client and server are using the following XML namespaces: http://schemas.microsoft.com/office/accessservices/2010/12/application

Individual **database applications** created using the **CreateApplication** method (section 3.1.4.1) MAY further restrict which protocol versions can be used when interacting with them. A protocol server can limit a database application to interact only with the **Access Services Protocol Version** with which it was created. Alternatively, the protocol server can change the set of **Access Services Protocol Versions** with which a database application will interact if there is a need to interact with protocol clients that support different **Access Services Protocol Versions** than the protocol client that created the database application. Operations specifying an **Access Services Protocol Version** which are being performed on database applications that do not support the given value MUST return a **SOAP fault** with the value "AppNotCompatibleWithServer" in the **Value** node of the **Subcode** element.

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
CreateApplication	This operation creates a new **database application** under the current **site**.
GetServerInformation	This operation retrieves information about the protocol server.

3.1.4.1 CreateApplication

This operation creates a new **database application** under the current **site**.

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

```
<wsdl:operation name="CreateApplication" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:CreateApplicationSoapIn"/>
  <wsdl:output message="tns:CreateApplicationSoapOut"/>
</wsdl:operation>
```

The client sends a **CreateApplicationSoapIn** (section 3.1.4.1.1.1) request message and the server responds with a **CreateApplicationSoapOut** (section 3.1.4.1.1.2) response message upon successful completion of creating the database application. The protocol server MUST respond with a **SOAP fault** if an error occurs on the protocol server during this operation.

3.1.4.1.1 Messages

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

Message	Description
CreateApplicationSoapIn	The request WSDL message for the **CreateApplication WSDL operation**.
CreateApplicationSoapOut	The response WSDL message for the **CreateApplication** WSDL operation.

3.1.4.1.1.1 CreateApplicationSoapIn

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

The **SOAP action** value is:

```
http://schemas.microsoft.com/office/Access/2010/11/Server/WebServices/ServerDesignService/CreateApplication
```

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

3.1.4.1.1.2 CreateApplicationSoapOut

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

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

3.1.4.1.2 Elements

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

Element	Description
CreateApplication	The input data for the **CreateApplication WSDL operation**.
CreateApplicationResponse	The result data for the **CreateApplication** WSDL operation.

3.1.4.1.2.1 CreateApplication

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

```
<xs:element name="CreateApplication" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="parameters"
type="tns:CreateApplicationParameters"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

parameters: A **CreateApplicationParameters** element (section 3.1.4.1.3.1) that specifies the information that the protocol server needs to create a new **database application**. MUST be present.

3.1.4.1.2.2 CreateApplicationResponse

The **CreateApplicationResponse** element specifies the result data for the **CreateApplication WSDL operation**.

```
<xs:element name="CreateApplicationResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="CreateApplicationResult"
type="tns:CreateAppResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

CreateApplicationResult: A **CreateAppResult** element (section 3.1.4.1.3.3) that specifies information about a newly created **database application** for the protocol client to use. MUST be present.

3.1.4.1.3 Complex Types

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

Complex type	Description
Collation	Specifies a **collation**.
CreateApplicationParameters	Specifies a **database application** to create.
CreateAppResult	Specifies information about the database application created.
ServiceParameters	Specifies values to a **CreateApplication** call.

3.1.4.1.3.1 CreateApplicationParameters

Namespace:
http://schemas.microsoft.com/office/Access/2010/11/Server/WebServices/ServerDesignService

Specifies the information that the protocol server needs in order to create a **database application**.

```
<xs:complexType name="CreateApplicationParameters"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexContent mixed="false">
    <xs:extension base="tns:ServiceParameters">
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="Collation" type="tns:Collation"/>
        <xs:element minOccurs="0" maxOccurs="1" name="ProtocolVersionOptions"
type="tns:ProtocolVersionList"/>
        <xs:element minOccurs="0" maxOccurs="1" name="AppProperties"
type="tns:AppPropertiesList"/>
      </xs:sequence>
      <xs:attribute name="Name" type="xs:string"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Collation: A **Collation** element (section 3.1.4.1.3.2) that specifies the **collation** to use in the new database application. MUST be present.

ProtocolVersionOptions: A **ProtocolVersionList** (section 2.2.4.1) element in which each **ProtocolVersion** element specifies an **Access Services Protocol Version** (section 3.1.1.1) that the protocol client supports. MUST be present.

AppProperties: MUST be ignored.

Name: A **string** ([XMLSCHEMA2] section 3.2.1) that specifies the name of the new database application. MUST be present.

3.1.4.1.3.2 Collation

Namespace:
http://schemas.microsoft.com/office/Access/2010/11/Server/WebServices/ServerDesignService

Specifies **collation** information.

```
<xs:complexType name="Collation" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:attribute name="Culture" type="xs:string"/>
  <xs:attribute name="CaseSensitive" type="xs:boolean"/>
  <xs:attribute name="AccentSensitive" type="xs:boolean"/>
  <xs:attribute name="KanaSensitive" type="xs:boolean"/>
  <xs:attribute name="WidthSensitive" type="xs:boolean"/>
  <xs:attribute name="UseSupplementaryCharacters" type="xs:boolean" use="required"/>
```

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

Culture: A **string** ([XMLSCHEMA2] section 3.2.1) that specifies the **culture name** of the collation. MUST be formatted as specified by [RFC4646]. MUST be present.

CaseSensitive: A **boolean** ([XMLSCHEMA2] section 3.2.2) that specifies whether the collation distinguishes between uppercase and lowercase letters. If omitted, the protocol server uses a default selection based on the **Culture** attribute.

AccentSensitive: A **boolean** ([XMLSCHEMA2] section 3.2.2) that specifies whether the collation distinguishes between characters with accents and those without. If omitted, the protocol server uses a default selection based on the **Culture** attribute.

KanaSensitive: A **boolean** ([XMLSCHEMA2] section 3.2.2) that specifies whether the collation distinguishes between the two types of Japanese kana characters: hiragana and katakana. If omitted, the protocol server uses a default selection based on the **Culture** attribute.

WidthSensitive: A **boolean** ([XMLSCHEMA2] section 3.2.2) that specifies whether the collation distinguishes between single-byte characters and the same characters when they are represented as double-byte characters. If omitted, the protocol server uses a default selection based on the **Culture** attribute.

UseSupplementaryCharacters: A **boolean** ([XMLSCHEMA2] section 3.2.2) that specifies whether the collation uses **surrogate pairs**.

3.1.4.1.3.3 CreateAppResult

Namespace:
http://schemas.microsoft.com/office/Access/2010/11/Server/WebServices/ServerDesignService

Specifies information about the **database application** that was created in response to the request from the protocol client.

```
<xs:complexType name="CreateAppResult" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexContent mixed="false">
    <xs:extension base="tns:ServiceResult">
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="ApplicationUrl" type="xs:string"/>
      </xs:sequence>
      <xs:attribute name="ProtocolVersion" type="xs:string"/>
      <xs:attribute name="AlternateUrl" type="xs:string"/>
      <xs:attribute name="ApplicationCulture" type="xs:string"/>
      <xs:attribute name="DatabaseName" type="xs:string"/>
      <xs:attribute name="DataServerName" type="xs:string"/>
      <xs:attribute name="RequestIPAddress" type="xs:string"/>
      <xs:attribute name="IsFirewallRestricted" type="xs:boolean" use="required"/>
      <xs:attribute name="IsDataConnectivitySupported" type="xs:boolean" use="required"/>
      <xs:attribute name="IsSendEmailSupported" type="xs:boolean" use="required"/>
      <xs:attribute name="PackageAppTitle" type="xs:string"/>
      <xs:attribute name="PackageAppVersion" type="xs:string"/>
      <xs:attribute name="PackageIsLocked" type="xs:boolean" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

ApplicationUrl: A **string** ([XMLSCHEMA2] section 3.2.1) that specifies the **fully qualified URL** to the database application that was successfully created. MUST be present.

ProtocolVersion: An **Access Services Protocol Version** (section 3.1.1.1) that specifies the best match between the versions supported by the protocol client and the protocol server, as determined

by the protocol server implementation. MUST be one of the values in the **ProtocolVersionOptions** of the **CreateApplicationParameters** (section 3.1.4.1.3.1) that is sent with the request from the protocol client. MUST also be one of the values that would be returned by the protocol server from a call to **GetServerInformation** (section 3.1.4.2) in the **AcceptedProtocolVersions** of a **GetServerInfoResult** (section 3.1.4.2.3.1).

AlternateUrl: A **string** ([XMLSCHEMA2] section 3.2.1) that specifies an alternate fully qualified URL at which the application can be accessed. The alternate **URL** MAY be a friendly or shortened version of the application's URL. MUST be present.

ApplicationCulture: A **string** ([XMLSCHEMA2] section 3.2.1) attribute that specifies the **culture name** of the application. MUST be formatted as specified by [RFC4646]. MUST be present.

DatabaseName: A **string** ([XMLSCHEMA2] section 3.2.1) attribute that specifies the name of the backend database that contains the application. The database is located on the server specified by **DataServerName**. MUST be present.

DataServerName: A **string** ([XMLSCHEMA2] section 3.2.1) attribute that specifies the name of the server on which the database specified by **DatabaseName** resides. MUST be present.

RequestIPAddress: A **string** ([XMLSCHEMA2] section 3.2.1) attribute that specifies the IP address from which the protocol server received the **CreateApplication** request. MUST be either an **IPv6 address in string format** ([RFC4291] section 2.2) or an **IPv4 address in string format** ([RFC1123] section 2.1). MUST be present.

IsFirewallRestricted: A **boolean** ([XMLSCHEMA2] section 3.2.2) attribute that specifies whether connections to the application are restricted by a firewall on the server. MUST be present.

IsDataConnectivitySupported: A **boolean** ([XMLSCHEMA2] section 3.2.2) attribute that specifies whether external connections to the application are enabled. MUST be present.

IsSendEmailSupported: A **boolean** ([XMLSCHEMA2] section 3.2.2) attribute that specifies whether the SendEmail data macro action is enabled in the application. MUST be present.

PackageAppTitle: A **string** ([XMLSCHEMA2] section 3.2.1) attribute that specifies the title of the application package. MUST be present. See description of **PackageAppTitle** ([MS-AADT] section 3.1.4.4.3.2).

PackageAppVersion: A **string** ([XMLSCHEMA2] section 3.2.1) that specifies the version of the application package. MUST be present. See description of **PackageAppVersion** ([MS-AADT] section 3.1.4.4.3.2).

PackageIsLocked: A **boolean** ([XMLSCHEMA2] section 3.2.2) that specifies whether customization of the application package is disabled. MUST be present. See description of **PackageIsLocked** ([MS-AADT] section 3.1.4.4.3.2).

3.1.4.1.3.4 ServiceParameters

Namespace:
http://schemas.microsoft.com/office/Access/2010/11/Server/WebServices/ServerDesignService

A base complex type extended by **CreateApplicationParameters** (section 3.1.4.1.3.1).

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

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 GetServerInformation

This operation retrieves information about the protocol server.

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

```
<wsdl:operation name="GetServerInformation" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:GetServerInformationSoapIn"/>
  <wsdl:output message="tns:GetServerInformationSoapOut"/>
</wsdl:operation>
```

The client sends a **GetServerInformationSoapIn** (section 3.1.4.2.1.1) request message and the server responds with a **GetServerInformationSoapOut** (section 3.1.4.2.1.2) response message upon successful completion. The protocol server MUST respond with a **SOAP fault** if an error occurs on the protocol server during this operation.

3.1.4.2.1 Messages

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

Message	Description
GetServerInformationSoapIn	The request WSDL message for the **GetServerInformation WSDL operation**.
GetServerInformationSoapOut	The response WSDL message for the **GetServerInformation** WSDL operation.

3.1.4.2.1.1 GetServerInformationSoapIn

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

The **SOAP action** value is:

```
http://schemas.microsoft.com/office/Access/2010/11/Server/WebServices/ServerDesignService/GetServerInformation
```

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

3.1.4.2.1.2 GetServerInformationSoapOut

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

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

3.1.4.2.2 Elements

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

Element	Description
GetServerInformation	The input data for the **GetServerInformation WSDL operation**.
GetServerInformationResponse	The result data for the **GetServerInformation** WSDL operation.

3.1.4.2.2.1 GetServerInformation

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

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

3.1.4.2.2.2 GetServerInformationResponse

The **GetServerInformationResponse** element specifies the result data for the **GetServerInformation WSDL operation**.

```
<xs:element name="GetServerInformationResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="GetServerInformationResult"
type="tns:GetServerInfoResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

GetServerInformationResult: A **GetServerInfoResult** (section 3.1.4.2.3.1) that specifies information about the protocol server. 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
GetServerInfoResult	Specifies information about a protocol server.

3.1.4.2.3.1 GetServerInfoResult

Namespace:
http://schemas.microsoft.com/office/Access/2010/11/Server/WebServices/ServerDesignService

Specifies information about the protocol server.

```
<xs:complexType name="GetServerInfoResult" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexContent mixed="false">
    <xs:extension base="tns:ServiceResult">
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="AcceptedProtocolVersions"
type="tns:ProtocolVersionList"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

AcceptedProtocolVersions: A **ProtocolVersionList** (section 2.2.4.1) element in which each **ProtocolVersion** element specifies an **Access Services Protocol Version** (section 3.1.1.1) that the protocol server supports. MUST be present.

3.1.4.2.4 Simple Types

None.

3.1.4.2.5 Attributes

None.

3.1.4.2.6 Groups

None.

3.1.4.2.7 Attribute Groups

None.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4 Protocol Examples

4.1 Use CreateApplication to create a database application

This example describes how to use **CreateApplication**, as described in section 3.1.4.1, to create a new **database application**.

The protocol client sends the following message to the protocol server using an **HTTP** POST to http://www.example.com/_vti_bin/accsvc/ServerDesignService.asmx to create a new database application called "exampleapplication":

```
<?xml version="1.0" encoding="UTF-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <CreateApplication
xmlns="http://schemas.microsoft.com/office/Access/2010/11/Server/WebServices/ServerDesignServ
ice">
      <parameters Name="exampleapplication" Template="">
        <Collation Culture="en-US" CaseSensitive="false" AccentSensitive="true"
KanaSensitive="false" WidthSensitive="false" UseSupplementaryCharacters="false" />
        <ProtocolVersionOptions>
          <ProtocolVersion>15.0.24.0</ProtocolVersion>
        </ProtocolVersionOptions>
      </parameters>
    </CreateApplication>
  </soap12:Body>
</soap12:Envelope>
```

In this example, the protocol client requests the protocol server to create an English (US) database application with a **collation** that is case insensitive and accent sensitive.

The protocol server responds with the following message:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <CreateApplicationResponse
xmlns="http://schemas.microsoft.com/office/Access/2010/11/Server/WebServices/ServerDesignServ
ice">
      <CreateApplicationResult ProtocolVersion="15.0.24.0">
        <Warning xsi:nil="true" />
        <ApplicationUrl>http://www.example.com/exampleapplication</ApplicationUrl>
      </CreateApplicationResult>
    </CreateApplicationResponse>
  </soap:Body>
</soap:Envelope>
```

In this example, the **ApplicationUrl** element contains the **fully qualified URL** of the successfully created database application.

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:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:tns="http://schemas.microsoft.com/office/Access/2010/11/Server/WebServices/ServerDesign
Service" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://schemas.microsoft.com/office/Access/2010/11/Server/WebServices/Server
DesignService" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <xs:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/office/Access/2010/11/Server/WebServices/Server
DesignService">
      <xs:element name="GetServerInformation">
        <xs:complexType/>
      </xs:element>
      <xs:element name="GetServerInformationResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="GetServerInformationResult"
type="tns:GetServerInfoResult"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="GetServerInfoResult">
        <xs:complexContent mixed="false">
          <xs:extension base="tns:ServiceResult">
            <xs:sequence>
              <xs:element minOccurs="0" maxOccurs="1" name="AcceptedProtocolVersions"
type="tns:ProtocolVersionList"/>
            </xs:sequence>
          </xs:extension>
        </xs:complexContent>
      </xs:complexType>
      <xs:complexType name="ServiceResult">
        <xs:sequence>
          <xs:element minOccurs="1" maxOccurs="1" name="Warning" nillable="true"
type="tns:ServiceWarning"/>
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="ServiceWarning">
        <xs:attribute name="Code" type="xs:string"/>
        <xs:attribute name="Message" type="xs:string"/>
      </xs:complexType>
      <xs:complexType name="ProtocolVersionList">
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="unbounded" name="ProtocolVersion"
type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
      <xs:element name="CreateApplication">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="parameters"
type="tns:CreateApplicationParameters"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="CreateApplicationParameters">
        <xs:complexContent mixed="false">
          <xs:extension base="tns:ServiceParameters">
            <xs:sequence>
              <xs:element minOccurs="0" maxOccurs="1" name="Collation" type="tns:Collation"/>
```

```
              <xs:element minOccurs="0" maxOccurs="1" name="ProtocolVersionOptions"
type="tns:ProtocolVersionList"/>

              <xs:element minOccurs="0" maxOccurs="1" name="AppProperties"
type="tns:AppPropertiesList"/>
            </xs:sequence>
            <xs:attribute name="Name" type="xs:string"/>
          </xs:extension>
        </xs:complexContent>
      </xs:complexType>
      <xs:complexType name="ServiceParameters"/>
      <xs:complexType name="Collation">
        <xs:attribute name="Culture" type="xs:string"/>
        <xs:attribute name="CaseSensitive" type="xs:boolean"/>
        <xs:attribute name="AccentSensitive" type="xs:boolean"/>
        <xs:attribute name="KanaSensitive" type="xs:boolean"/>
        <xs:attribute name="WidthSensitive" type="xs:boolean"/>
        <xs:attribute name="UseSupplementaryCharacters" type="xs:boolean" use="required"/>
      </xs:complexType>
      <xs:complexType name="AppPropertiesList">
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="unbounded" name="AppProperty"
type="tns:AppProperty"/>
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="AppProperty">
        <xs:attribute name="Name" type="xs:string"/>
        <xs:attribute name="Value" type="xs:string"/>
      </xs:complexType>
      <xs:element name="CreateApplicationResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="CreateApplicationResult"
type="tns:CreateAppResult"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="CreateAppResult">
        <xs:complexContent mixed="false">
          <xs:extension base="tns:ServiceResult">
            <xs:sequence>
              <xs:element minOccurs="0" maxOccurs="1" name="ApplicationUrl"
type="xs:string"/>
            </xs:sequence>
            <xs:attribute name="ProtocolVersion" type="xs:string"/>
            <xs:attribute name="AlternateUrl" type="xs:string"/>
            <xs:attribute name="ApplicationCulture" type="xs:string"/>
            <xs:attribute name="DatabaseName" type="xs:string"/>
            <xs:attribute name="DataServerName" type="xs:string"/>
            <xs:attribute name="RequestIPAddress" type="xs:string"/>
            <xs:attribute name="IsFirewallRestricted" type="xs:boolean" use="required"/>
            <xs:attribute name="IsDataConnectivitySupported" type="xs:boolean"
use="required"/>
            <xs:attribute name="IsSendEmailSupported" type="xs:boolean" use="required"/>
            <xs:attribute name="PackageAppTitle" type="xs:string"/>
            <xs:attribute name="PackageAppVersion" type="xs:string"/>
            <xs:attribute name="PackageIsLocked" type="xs:boolean" use="required"/>
          </xs:extension>
        </xs:complexContent>
      </xs:complexType>
    </xs:schema>
  </wsdl:types>
  <wsdl:portType name="ServerDesignServiceSoap">
    <wsdl:operation name="GetServerInformation">
      <wsdl:input message="tns:GetServerInformationSoapIn"/>
      <wsdl:output message="tns:GetServerInformationSoapOut"/>
    </wsdl:operation>
    <wsdl:operation name="CreateApplication">
      <wsdl:input message="tns:CreateApplicationSoapIn"/>
```

```
      <wsdl:output message="tns:CreateApplicationSoapOut"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="ServerDesignServiceSoap" type="tns:ServerDesignServiceSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="GetServerInformation">
      <soap:operation
soapAction="http://schemas.microsoft.com/office/Access/2010/11/Server/WebServices/ServerDesig
nService/GetServerInformation" style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="CreateApplication">
      <soap:operation
soapAction="http://schemas.microsoft.com/office/Access/2010/11/Server/WebServices/ServerDesig
nService/CreateApplication" style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:binding name="ServerDesignServiceSoap12" type="tns:ServerDesignServiceSoap">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="GetServerInformation">
      <soap12:operation
soapAction="http://schemas.microsoft.com/office/Access/2010/11/Server/WebServices/ServerDesig
nService/GetServerInformation" style="document"/>
      <wsdl:input>
        <soap12:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="CreateApplication">
      <soap12:operation
soapAction="http://schemas.microsoft.com/office/Access/2010/11/Server/WebServices/ServerDesig
nService/CreateApplication" style="document"/>
      <wsdl:input>
        <soap12:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:message name="CreateApplicationSoapIn">
    <wsdl:part name="parameters" element="tns:CreateApplication"/>
  </wsdl:message>
  <wsdl:message name="CreateApplicationSoapOut">
    <wsdl:part name="parameters" element="tns:CreateApplicationResponse"/>
  </wsdl:message>
  <wsdl:message name="GetServerInformationSoapIn">
    <wsdl:part name="parameters" element="tns:GetServerInformation"/>
  </wsdl:message>
  <wsdl:message name="GetServerInformationSoapOut">
    <wsdl:part name="parameters" element="tns:GetServerInformationResponse"/>
  </wsdl:message>
</wsdl:definitions>
```

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 Access 2013
- Microsoft SharePoint Server 2013
- Microsoft Access 2016
- Microsoft SharePoint Server 2016
- Microsoft Access 2019
- Microsoft SharePoint Server 2019
- Microsoft Access 2021

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

T

U

V

W