

[MS-ASDT]: Access Server Design Time Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
01/20/2012	0.1	New	Released new document.
04/11/2012	0.1	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	0.1	No change	No changes to the meaning, language, or formatting of the technical content.
09/12/2012	0.1	No change	No changes to the meaning, language, or formatting of the technical content.
10/08/2012	1.0	Major	Significantly changed the technical content.
02/11/2013	1.0	No change	No changes to the meaning, language, or formatting of the technical content.
07/30/2013	1.0	No change	No changes to the meaning, language, or formatting of the technical content.
11/18/2013	1.0	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction	5
1.1 Glossary	5
1.2 References	6
1.2.1 Normative References	6
1.2.2 Informative References	7
1.3 Overview	7
1.4 Relationship to Other Protocols	7
1.5 Prerequisites/Preconditions	7
1.6 Applicability Statement	7
1.7 Versioning and Capability Negotiation	8
1.8 Vendor-Extensible Fields	8
1.9 Standards Assignments	8
2 Messages	9
2.1 Transport	9
2.2 Common Message Syntax	9
2.2.1 Namespaces	9
2.2.2 Messages	9
2.2.3 Elements	9
2.2.4 Complex Types	10
2.2.4.1 ProtocolVersionList	10
2.2.4.2 ServiceResult	10
2.2.4.3 ServiceWarning	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 Server Details	12
3.1.1 Abstract Data Model	12
3.1.1.1 Access Services Protocol Version	12
3.1.2 Timers	12
3.1.3 Initialization	12
3.1.4 Message Processing Events and Sequencing Rules	13
3.1.4.1 CreateApplication	13
3.1.4.1.1 Messages	13
3.1.4.1.1.1 CreateApplicationSoapIn	13
3.1.4.1.1.2 CreateApplicationSoapOut	13
3.1.4.1.2 Elements	14
3.1.4.1.2.1 CreateApplication	14
3.1.4.1.2.2 CreateApplicationResponse	14
3.1.4.1.3 Complex Types	14
3.1.4.1.3.1 CreateApplicationParameters	15
3.1.4.1.3.2 Collation	15
3.1.4.1.3.3 CreateAppResult	16
3.1.4.1.3.4 ServiceParameters	17
3.1.4.1.4 Simple Types	17
3.1.4.1.5 Attributes	17
3.1.4.1.6 Groups	17

3.1.4.1.7	Attribute Groups	17
3.1.4.2	GetServerInformation	18
3.1.4.2.1	Messages	18
3.1.4.2.1.1	GetServerInformationSoapIn	18
3.1.4.2.1.2	GetServerInformationSoapOut	18
3.1.4.2.2	Elements.....	18
3.1.4.2.2.1	GetServerInformation	19
3.1.4.2.2.2	GetServerInformationResponse	19
3.1.4.2.3	Complex Types	19
3.1.4.2.3.1	GetServerInfoResult	19
3.1.4.2.4	Simple Types.....	20
3.1.4.2.5	Attributes.....	20
3.1.4.2.6	Groups.....	20
3.1.4.2.7	Attribute Groups	20
3.1.5	Timer Events	20
3.1.6	Other Local Events	20
4	Protocol Examples.....	21
4.1	Use CreateApplication to create a database application.....	21
5	Security.....	22
5.1	Security Considerations for Implementers.....	22
5.2	Index of Security Parameters	22
6	Appendix A: Full WSDL.....	23
7	Appendix B: Product Behavior	27
8	Change Tracking.....	28
9	Index	29

1 Introduction

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

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in RFC 2119. Sections 1.5 and 1.9 are also normative but cannot contain those terms. All other sections and examples in this specification are informative.

1.1 Glossary

The following terms are defined in [\[MS-GLOS\]](#):

Hypertext Transfer Protocol (HTTP)
Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)
IPv4 address in string format
IPv6 address in string format
SOAP
SOAP action
SOAP body
SOAP fault
XML namespace

The following terms are defined in [\[MS-OFCSGLOS\]](#):

collation
culture name
database application
endpoint
fully qualified URL
site
SOAP fault code
Uniform Resource Identifier (URI)
Uniform Resource Locator (URL)
Web Services Description Language (WSDL)
WSDL message
WSDL operation
XML namespace prefix
XML schema

The following terms are specific to this document:

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].

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

1.2 References

References to Microsoft Open Specifications documentation do not include a publishing year because links are to the latest version of the documents, which are updated frequently. References to other documents include a publishing year when one is available.

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, <http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624>, as an additional source.

[RFC1123] Braden, R., "Requirements for Internet Hosts - Application and Support", STD 3, 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.ietf.org/rfc/rfc2616.txt>

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000, <http://www.ietf.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] A. Phillips, Ed., and M. Davis, Ed., "Tags for Identifying Languages", BCP 47, RFC 4646, September 2006, <http://www.ietf.org/rfc/rfc4646.txt>

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

[SOAP1.2/1] Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J., and Nielsen, H.F., "SOAP Version 1.2 Part 1: Messaging Framework", W3C Recommendation, June 2003, <http://www.w3.org/TR/2003/REC-soap12-part1-20030624>

[SOAP1.2/2] Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J., and Nielsen, H.F., "SOAP Version 1.2 Part 2: Adjuncts", W3C Recommendation, June 2003, <http://www.w3.org/TR/2003/REC-soap12-part2-20030624>

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

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

[XMLSCHEMA1] Thompson, H.S., Beech, D., Maloney, M., Eds., and Mendelsohn, N., Ed., "XML Schema Part 1: Structures", W3C Recommendation, May 2001, <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>

[XMLSCHEMA2] Biron, P.V., and Malhotra, A., Eds., "XML Schema Part 2: Datatypes", W3C Recommendation, May 2001, <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>

1.2.2 Informative References

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

[MS-GLOS] Microsoft Corporation, "[Windows Protocols Master Glossary](#)".

[MS-OFCGLOS] Microsoft Corporation, "[Microsoft Office Master Glossary](#)".

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:

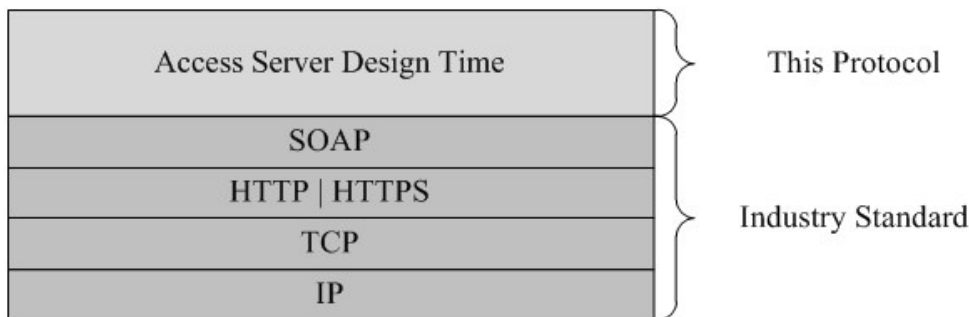


Figure 1: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

This protocol operates against a **site (2)** that is identified by a **URI** that is known by protocol clients. The protocol server **endpoint (4)** is formed by appending `"/_vti_bin/acccsvc/ServerDesignService.asmx"` to the URI of the site (2), for example: `http://www.example.com/Repository/_vti_bin/acccsvc/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 [\[SOAP1.2/1\]](#) and [\[SOAP1.2/2\]](#), 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/wsd/http/	
soap	http://schemas.xmlsoap.org/wsd/soap/	[SOAP1.1]
soap12	http://schemas.xmlsoap.org/wsd/soap12/	[SOAP1.2/1] [SOAP1.2/2]
tns	http://schemas.microsoft.com/office/Access/2010/11/Server/WebServices/ServerDesignService	
wSDL	http://schemas.xmlsoap.org/wsd/	[WSDL]
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1] [XMLSCHEMA2]

2.2.2 Messages

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

2.2.3 Elements

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

2.2.4 Complex Types

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

Complex type	Description
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 (2).
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 (2).

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/Cre
ateApplication
```

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 .

Complex type	Description
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: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.

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: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.

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/ServerDesignService">
      <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/ServerDesignService">
      <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"?>
<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:schema>
  </wsdl:types>

```

```

    </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: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: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: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/ServerDesignService/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/ServerDesignService/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/ServerDesignService/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/ServerDesignService/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 released service packs:

- Microsoft Access 2013
- Microsoft SharePoint Server 2013

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

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
[Access services ProtocolVersion token](#) 12
[Applicability](#) 7
[Attribute groups](#) 11
[Attributes](#) 11

C

[Capability negotiation](#) 8
[Change tracking](#) 28
[Complex types](#) 10
[ProtocolVersionList](#) 10
[ServiceResult](#) 10
[ServiceWarning](#) 11

D

Data model - abstract
[server](#) 12
[Access services ProtocolVersion token](#) 12

E

Events
[local - server](#) 20
[timer - server](#) 20
Examples
[use CreateApplication to create a database application](#) 21

F

[Fields - vendor-extensible](#) 8
[Full WSDL](#) 23

G

[Glossary](#) 5
[Groups](#) 11

I

[Implementer - security considerations](#) 22
[Index of security parameters](#) 22
[Informative references](#) 7
Initialization
[server](#) 12
[Introduction](#) 5

L

Local events
[server](#) 20

M

Message processing
[server](#) 13
Messages
[attribute groups](#) 11
[attributes](#) 11
[complex types](#) 10
[elements](#) 9
[enumerated](#) 9
[groups](#) 11
[namespaces](#) 9
[ProtocolVersionListcomplex type](#) 10
[ServiceResultcomplex type](#) 10
[ServiceWarningcomplex type](#) 11
[simple types](#) 11
[syntax](#) 9
[transport](#) 9

N

[Namespaces](#) 9
[Normative references](#) 6

O

Operations
[CreateApplication](#) 13
[GetServerInformation](#) 18
[Overview \(synopsis\)](#) 7

P

[Parameters - security index](#) 22
[Preconditions](#) 7
[Prerequisites](#) 7
[Product behavior](#) 27
Protocol Details
[overview](#) 12
[ProtocolVersion token](#) 12
[ProtocolVersionListcomplex type](#) 10

R

[References](#) 6
[informative](#) 7
[normative](#) 6
[Relationship to other protocols](#) 7

S

Security
[implementer considerations](#) 22
[parameter index](#) 22
Sequencing rules
[server](#) 13
Server
[abstract data model](#) 12
[Access services ProtocolVersion token](#) 12
[CreateApplication operation](#) 13
[GetServerInformation operation](#) 18

- [initialization](#) 12
- [local events](#) 20
- [message processing](#) 13
- [sequencing rules](#) 13
- [timer events](#) 20
- [timers](#) 12
- [ServiceResultcomplex type](#) 10
- [ServiceWarningcomplex type](#) 11
- [Simple types](#) 11
- [Standards assignments](#) 8
- Syntax
 - [messages - overview](#) 9

T

- Timer events
 - [server](#) 20
- Timers
 - [server](#) 12
- [Tracking changes](#) 28
- [Transport](#) 9
- Types
 - [complex](#) 10
 - [simple](#) 11

U

- [Use CreateApplication to create a database application example](#) 21

V

- [Vendor-extensible fields](#) 8
- [Versioning](#) 8

W

- [WSDL](#) 23