

[MS-ART]:

Access Run 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
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	2.0	Major	Significantly changed the technical content.
7/30/2013	2.0	None	No changes to the meaning, language, or formatting of the technical content.
11/18/2013	2.0	None	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	2.0	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	2.0	None	No changes to the meaning, language, or formatting of the technical content.
7/31/2014	2.0	None	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	2.0	None	No changes to the meaning, language, or formatting of the technical content.
2/26/2016	3.0	Major	Significantly changed 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	6
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	Message Syntax.....	9
2.2.1	Complex Types.....	10
2.2.1.1	ClientMessage	11
2.2.1.2	CurrentUserPermissions	11
2.2.1.3	FieldSchema.....	12
2.2.1.4	FilterInfo.....	13
2.2.1.5	FixupRecord	13
2.2.1.6	FormatInfo.....	14
2.2.1.7	PagingInfo	15
2.2.1.8	ParameterValue	17
2.2.1.9	RecordSet	17
2.2.1.10	ServiceError	18
2.2.1.11	ServiceResult	19
2.2.1.12	SharedDataBaseInfo.....	19
2.2.1.13	UpdateRecord.....	22
2.2.2	Simple Types	22
2.2.2.1	CacheCommands	23
2.2.2.2	ClientMessageID	23
2.2.2.3	MessageSeverity.....	25
3	Protocol Details.....	27
3.1	Server Details.....	27
3.1.1	Abstract Data Model.....	27
3.1.2	Timers	27
3.1.3	Initialization.....	27
3.1.4	Higher-Layer Triggered Events	27
3.1.5	Message Processing Events and Sequencing Rules	27
3.1.5.1	AccessPortal.....	28
3.1.5.1.1	GetData	28
3.1.5.1.1.1	Request Body	29
3.1.5.1.1.2	Response Body.....	29
3.1.5.1.1.3	Processing Details.....	29
3.1.5.1.2	InsertRecords	30
3.1.5.1.2.1	Request Body	30
3.1.5.1.2.2	Response Body.....	31
3.1.5.1.2.3	Processing Details.....	31
3.1.5.1.3	UpdateRecords.....	32
3.1.5.1.3.1	Request Body	32
3.1.5.1.3.2	Response Body.....	33
3.1.5.1.3.3	Processing Details.....	33

3.1.5.1.4	DeleteRecords.....	34
3.1.5.1.4.1	Request Body	34
3.1.5.1.4.2	Response Body	34
3.1.5.1.4.3	Processing Details	35
3.1.5.1.5	GetDistinctValues	35
3.1.5.1.5.1	Request Body	35
3.1.5.1.5.2	Response Body	36
3.1.5.1.5.3	Processing Details	36
3.1.5.1.6	FixupRow	36
3.1.5.1.6.1	Request Body	37
3.1.5.1.6.2	Response Body	37
3.1.5.1.6.3	Processing Details	38
3.1.5.1.7	GetSearchData.....	38
3.1.5.1.7.1	Request Body	38
3.1.5.1.7.2	Response Body	39
3.1.5.1.7.3	Processing Details	39
3.1.6	Timer Events.....	40
3.1.6.1	Session Timeout	40
3.1.7	Other Local Events.....	40
4	Protocol Examples	41
4.1	GetData Service Operation	41
4.2	InsertRecords Service Operation	43
4.3	UpdateRecords Service Operation	45
4.4	DeleteRecords Service Operation	47
5	Security	50
5.1	Security Considerations for Implementers	50
5.2	Index of Security Parameters	50
6	Appendix A: Full JSON ABNF.....	51
7	Appendix B: Product Behavior	57
8	Change Tracking.....	58
9	Index.....	60

1 Introduction

The Access Run Time Protocol enables a protocol client to read, insert, update or delete data in a **database application**.

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:

alert: A message that is passed to a protocol client to notify it when specific criteria are met.

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

caption: One or more characters that can be used as a label for display purposes or as an identifier.

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

field: A discrete unit of a record that has a name, a data type, and a value.

Help topic identifier: A unique identifier for an article that contains Help content.

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 [\[RFC5246\]](#).

JavaScript Object Notation (JSON): A text-based, data interchange format that is used to transmit structured data, typically in Asynchronous JavaScript + XML (AJAX) web applications, as described in [\[RFC4627\]](#). The JSON format is based on the structure of ECMAScript (Jscript, JavaScript) objects.

primary key: A field or set of fields that uniquely identifies each record in a table. A primary key cannot contain a null value.

record: A group of related **fields**, which are sometimes referred to as columns, of information that are treated as a unit. Also referred to as row.

Request-URI: A **URI** in an **HTTP** request message, as described in [\[RFC2616\]](#).

result set: A list of records that results from running a stored procedure or query, or applying a filter. The structure and content of the data in a result set varies according to the implementation.

row: A collection of columns (1) that contains property values that describe a single item in a set of items that match the restriction (1) specified in a query.

session: A representation of application data in system memory. It is used to maintain state for application data that is being manipulated or monitored on a protocol server by a user.

session identifier: A key that enables an application to make reference to a session.

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.

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 [\[RFC1738\]](#).

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-ADR] Microsoft Corporation, "[Access Services Data Run Time Protocol](#)".

[MS-AXL2] Microsoft Corporation, "[Access Application Transfer Data Structure Version 2](#)".

[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>

[RFC4627] Crockford, D., "The application/json Media Type for JavaScript Object Notation (JSON)", RFC 4627, July 2006, <http://www.ietf.org/rfc/rfc4627.txt>

[RFC5234] Crocker, D., Ed., and Overell, P., "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008, <http://www.rfc-editor.org/rfc/rfc5234.txt>

1.2.2 Informative References

None.

1.3 Overview

This protocol is used for manipulating data maintained by a protocol server. Communication is always initiated by the protocol client using different operations, the functionality of which is outlined following.

- Inserting, reading, updating and deleting data from the database application.
- Filtering, sorting data while reading it from the database application.
- Modeling hypothetical updates on the protocol server.
- Reading distinct values for a certain **field** in a **result set**.

A typical scenario for this protocol is opening a result set and updating or deleting data.

1.4 Relationship to Other Protocols

This protocol enables a protocol client to send a request that calls methods and accesses data on a protocol server, and then receive a corresponding response from the protocol server. This protocol depends on other structures and protocols to transport messages. Applications are layered on top of this protocol and they interact directly with this protocol specification.

The messages that are sent from the protocol client to the protocol server are formatted as **JSON**. It transmits those messages by using **HTTP**, as described in [\[RFC2616\]](#), or **HTTPS**, as described in [\[RFC2818\]](#). Responses from the protocol server are formatted as JSON.

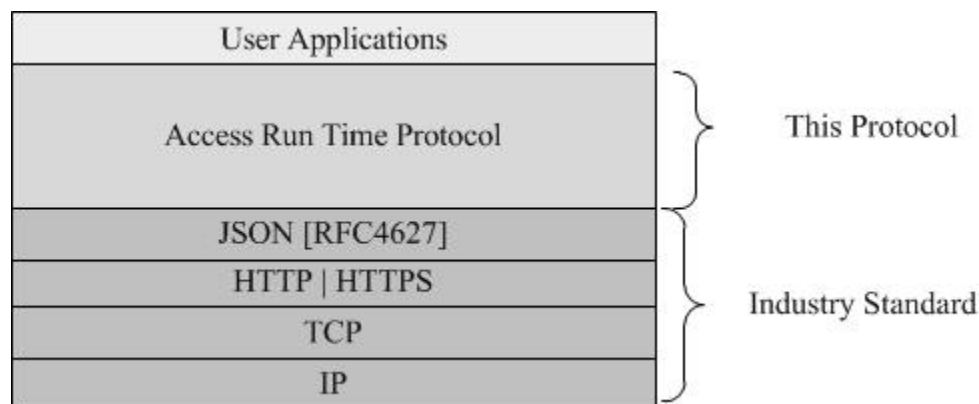


Figure 1 This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

This protocol operates against a protocol server that is configured to listen for HTTP or HTTPS requests and a protocol client that knows the **Request-URI** of the protocol server.

1.6 Applicability Statement

This protocol is applicable for the following scenarios:

- Inserting, reading, updating, or deleting data from the database application.
- Retrieving data from the database application in **Pages** (section [3.1.1](#)). This applies only if the number of **records** exceeds a threshold specified by the protocol server.
- Filtering data by retrieving only the data matching given restrictions.

- Sorting data retrieved from the database application
- Retrieving only distinct values for certain fields in the result set from the database application
- Modeling hypothetical updates on the protocol server.

This protocol is not applicable for the following scenarios:

- Creating, updating or deleting objects on the database application.
- Acting as a data transfer channel between protocol client and protocol server. Instead the data sent to protocol client is intended to be rendered for end user viewing.

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the area of supported transports. This protocol can use HTTP or HTTPS as a transport. For more information, see Transport (section [2.1](#)).

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

Protocol servers MUST support HTTP, as specified in [\[RFC2616\]](#). Protocol servers SHOULD also additionally support HTTPS, as specified in [\[RFC2818\]](#), to help secure connections with protocol clients.

Protocol messages MUST be formatted as JSON as specified in [\[RFC4627\]](#). Protocol clients MUST use the **GET** ([\[RFC2616\]](#) section 9.3) or **POST** ([\[RFC2616\]](#) section 9.5) method to send messages to the protocol servers.

2.2 Message Syntax

This section contains common definitions used by this protocol specification. The syntax of the definitions uses JavaScript Object Notation (JSON), as specified in [\[RFC4627\]](#), and the common JSON in **ABNF**, as specified by [\[RFC5234\]](#).

```
json-begin-array = json-whitespace %x5B json-whitespace ; [ left square bracket
json-begin-object = json-whitespace %x7B json-whitespace ; { left curly bracket
json-end-array = json-whitespace %x5D json-whitespace ; ] right square bracket
json-end-object = json-whitespace %x7D json-whitespace ; } right curly bracket
json-name-separator = json-whitespace %x3A json-whitespace ; : colon
json-value-separator = json-whitespace %x2C json-whitespace ; , comma
json-whitespace = *(
    %x20 / ; Space
    %x09 / ; Horizontal tab
    %x0A / ; Line feed or New line
    %x0D / ; Carriage return
)
json-value = json-false / json-null / json-true / json-object / json-array / json-number /
json-string
json-false = %x66.61.6c.73.65 ; false
json-null = %x6e.75.6c.6c ; null
json-true = %x74.72.75.65 ; true
json-bool = json-false / json-true
json-object = json-begin-object [json-member *( json-value-separator json-member ) ] json-
end-object
json-member = json-string json-name-separator json-value
json-number = [ json-minus ] json-int [ json-frac ] [ json-exp ]
json-decimal-point = %x2E ; .
json-digit1-9 = %x31-39 ; 1-9
json-e = %x65 / %x45 ; e E
json-exp = json-e [ json-minus / json-plus ] 1*DIGIT
json-frac = json-decimal-point 1*DIGIT
json-int = json-zero / ( json-digit1-9 *DIGIT )
json-minus = %x2D ; -
json-plus = %x2B ; +
json-zero = %x30 ; 0
json-array = json-begin-array [ json-value *( json-value-separator json-value ) ] json-end-
array
json-string = json-quotation-mark *json-char json-quotation-mark
json-char = json-unescape /
    json-escape (
        %x22 / ; " quotation mark U+0022
        %x5C / ; \ reverse solidus U+005C
        %x2F / ; / solidus U+002F
        %x62 / ; b backspace U+0008
        %x66 / ; f form feed U+000C
        %x6E / ; n line feed U+000A
        %x72 / ; r carriage return U+000D
        %x74 / ; t tab U+0009
        %x75 4HEXDIG ) ; uXXXX U+XXXX
    json-escape = %x5C ; \
```

```

json-quotation-mark = %x22 ; "
json-unesescaped = %x20-21 / %x23-5B / %x5D-10FFFF

```

The following table defines the common JSON types used in this document.

Common Type	Value
JsonAnyValue	json-value
JsonStringValue	json-string
JsonArrayOfAnyType	json-array
JsonArrayOfString	json-begin-array [json-string *(json-value-separator json-string)] json-end-array
JsonArrayOfInt	json-begin-array [json-int *(json-value-separator json-int)] json-end-array
JsonArrayOfArrayOfAnyType	json-begin-array [JsonArrayOfAnyType *(json-value-separator JsonArrayOfAnyType)] json-end-array
JsonArrayOfArrayOfString	json-begin-array [JsonArrayOfString *(json-value-separator JsonArrayOfString)] json-end-array
JsonArrayOfArrayOfArrayOfString	json-begin-array [JsonArrayOfArrayOfString *(json-value-separator JsonArrayOfArrayString)] json-end-array

2.2.1 Complex Types

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

Complex Type	Description
ClientMessage	Specifies an error or informative message to return from the protocol server. See section 2.2.1.1 for more details.
FieldSchema	Specifies all metadata information about a field of a Source ([MS-ADR] section 3.1.1.1).

Complex Type	Description
	See section 2.2.1.3 for more details.
FilterInfo	Specifies how to filter a result set. See section 2.2.1.4 for more details.
FixupRecord	Specifies the values for the fields of the records that are to be used for the modeling a hypothetical update on the protocol server. See section 2.2.1.5 for more details.
FormatInfo	Specifies all metadata information about a field of a Source ([MS-ADR] section 3.1.1.1). See section 2.2.1.6 for more details.
PagingInfo	Specifies which records of data from the Source (section 3.1.1) are requested by the protocol client. See section 2.2.1.7 for more details.
ParameterValue	Specifies a container for the name of a CT_Parameter ([MS-AXL2] section 2.2.3.12) and its associated value. See section 2.2.1.8 for more details.
RecordSet	A complex type that is output by all the protocol operations. See section 2.2.1.9 for more details.
ServiceError	Specifies information about an alert (2) sent from protocol server to the protocol client. See section 2.2.1.10 for more details.
ServiceResult	Specifies the result of the protocol operation. See section 2.2.1.11 for more details.
SharedDataBaseInfo	Specifies information about the Source (section 3.1.1) on which protocol operations are to be performed. See section 2.2.1.12 for more details.
UpdateRecord	Specifies information about original values and new values of fields in a record that is to be inserted or updated in database application. See section 2.2.1.13 for more details.

2.2.1.1 ClientMessage

The meaning of **ClientMessage** is specified by [MS-ADR] section 2.2.4.24. The format of the **ClientMessage** complex type is specified by this section. The following ABNF references types specified by section [2.2](#).

```
ClientMessage = json-object
MessageID = json-quotation-mark "MessageID" json-quotation-mark json-name-
separator ClientMessageID
Context = json-quotation-mark "Context" json-quotation-mark json-name-separator
JsonArrayOfAnyType
```

The meanings of **Context** and **MessageID** are specified by [MS-ADR] section 2.2.4.24.

ClientMessage MUST contain exactly two **json-member** instances, and each instance MUST be different. Each instance MUST be one of the following: **Context** and **MessageID**.

2.2.1.2 CurrentUserPermissions

The meaning of **CurrentUserPermissions** is specified by [MS-ADR] section 3.1.4.45.3.11. The format of the **CurrentUserPermissions** complex type is specified by this section. The following ABNF references types specified by section [2.2](#).

```
CurrentUserPermissions = json-object
```

```
Read = json-quotation-mark "Read" json-quotation-mark json-name-separator json-bool
Write = json-quotation-mark "Write" json-quotation-mark json-name-separator json-bool
Author = json-quotation-mark "Author" json-quotation-mark json-name-separator json-bool
IsAuthenticated = json-quotation-mark "IsAuthenticated" json-quotation-mark json-name-separator json-bool
```

The meanings of **Read**, **Write**, **Author**, and **IsAuthenticated** are specified by [MS-ADR] section 3.1.4.45.3.11.

CurrentUserPermissions MUST contain exactly four **json-member** instances, and each instance MUST be different. Each instance MUST be one of the following: **Read**, **Write**, **Author**, and **IsAuthenticated**.

2.2.1.3 FieldSchema

The meaning of **FieldSchema** is specified by [MS-ADR] section 2.2.4.25. The format of the **FieldSchema** complex type is specified by this section. The following ABNF references types specified by section [2.2](#).

```
FieldSchema = json-object
ColumnName = json-quotation-mark "ColumnName" json-quotation-mark json-name-separator json-string
DataType = json-quotation-mark "DataType" json-quotation-mark json-name-separator json-string
DefaultValue = json-quotation-mark "DefaultValue" json-quotation-mark json-name-separator json-string
IsKey = json-quotation-mark "IsKey" json-quotation-mark json-name-separator json-bool
Required = json-quotation-mark "Required" json-quotation-mark json-name-separator json-bool
ReadOnly = json-quotation-mark "ReadOnly" json-quotation-mark json-name-separator json-bool
MaxLength = json-quotation-mark "MaxLength" json-quotation-mark json-name-separator json-int
DefaultExpression = json-quotation-mark "DefaultExpression" json-quotation-mark json-name-separator json-string
ValidationScript = json-quotation-mark "ValidationScript" json-quotation-mark json-name-separator json-string
ValidationMessage = json-quotation-mark "ValidationMessage" json-quotation-mark json-name-separator ClientMessage
KeyIndex = json-quotation-mark "KeyIndex" json-quotation-mark json-name-separator json-int
SourceObject = json-quotation-mark "SourceObject" json-quotation-mark json-name-separator json-string
DependentFields = json-quotation-mark "DependentFields" json-quotation-mark json-name-separator JsonArrayOfInt
AllowMultipleValues = json-quotation-mark "AllowMultipleValues" json-quotation-mark json-name-separator json-bool
FormatString = json-quotation-mark "FormatString" json-quotation-mark json-name-separator json-string
CurrencySymbol = json-quotation-mark "CurrencySymbol" json-quotation-mark json-name-separator json-string
DecimalPlaces = json-quotation-mark "DecimalPlaces" json-quotation-mark json-name-separator json-int
```

```

TextType = json-quotation-mark "TextType" json-quotation-mark json-name-
separator json-string
IsTableQueryLookup = json-quotation-mark "IsTableQueryLookup" json-quotation-mark
json-name-separator json-bool
LookupSource = json-quotation-mark "LookupSource" json-quotation-mark json-name-
separator json-string
LookupBoundField = json-quotation-mark "LookupBoundField" json-quotation-mark json-
name-separator json-string
LookupDisplayField = json-quotation-mark "LookupDisplayField" json-quotation-mark
json-name-separator json-string

```

The meanings of **ColumnName**, **DataType**, **DefaultValue**, **IsKey**, **Required**, **ReadOnly**, **MaxLength**, **DefaultExpression**, **ValidationScript**, **ValidationMessage**, **KeyIndex**, **SourceObject**, **DependentFields**, **AllowMultipleValues**, **FormatString**, **CurrencySymbol**, **DecimalPlaces**, **TextType**, **IsTableQueryLookup**, **LookupSource**, **LookupBoundField**, and **LookupDisplayField** are specified by [MS-ADR] section 2.2.4.25.

FieldSchema MUST contain at least one and no more than twenty two **json-member** instances, and each instance MUST be different. Each instance MUST be one of the following: **ColumnName**, **DataType**, **DefaultValue**, **IsKey**, **Required**, **ReadOnly**, **MaxLength**, **DefaultExpression**, **ValidationScript**, **ValidationMessage**, **KeyIndex**, **SourceObject**, **DependentFields**, **AllowMultipleValues**, **FormatString**, **CurrencySymbol**, **DecimalPlaces**, **TextType**, **IsTableQueryLookup**, **LookupSource**, **LookupBoundField**, and **LookupDisplayField**.

2.2.1.4 FilterInfo

The meaning of **FilterInfo** is specified by [MS-ADR] section 2.2.4.10. The format of the **FilterInfo** complex type is specified by this section. The following ABNF references types specified by section 2.2.

```

FilterInfo = json-object
Culture = json-quotation-mark "Culture" json-quotation-mark json-name-separator json-string
Expression = json-quotation-mark "Expression" json-quotation-mark json-name-separator json-
string
Fields = json-quotation-mark "Fields" json-quotation-mark json-name-separator
JsonArrayOfStrings
Text = json-quotation-mark "Text" json-quotation-mark json-name-separator json-string

```

The meanings of **Culture**, **Expression**, **Fields**, and **Text** are specified by [MS-ADR] section 2.2.4.10.

FilterInfo MUST contain at least one and no more than four **json-member** instances, and each instance MUST be different. Each instance MUST be one of the following: **Culture**, **Expression**, **Fields**, and **Text**. If one of the **json-member** instances is **Text**, then an instance of **Fields** and an instance of **Culture** MUST also be present.

2.2.1.5 FixupRecord

The **FixupRecord** type specifies the values for the fields of **Updatable Source** (section 3.1.1) which are used for modeling a hypothetical update on the protocol server. These values are only used for the hypothetical update, and are not updated in the database application.

The **FixupRecord** type is an input for the **FixupRow** operation (section 3.1.5.1.6).

The following ABNF references types specified by section 2.2 and applies to the **FixupRecord**, unless extended or overridden by an operation specification.

```

FixupRecord = json-object

```

```

Key = json-quotation-mark "Key" json-quotation-mark json-name-separator json-string
SupportingFieldIndexes = json-quotation-mark "SupportingFieldIndexes" json-quotation-
mark json-name-separator JsonArrayOfInt
SupportingFieldValues = json-quotation-mark "SupportingFieldValues" json-quotation-mark
json-name-separator JsonArrayOfString

```

FixupRecord MUST contain three **json-member** instances, and each instance MUST be different. Each instance MUST be one of the following: **Key**, **SupportingFieldIndexes** and **SupportingFieldValues**.

Key: A **json-string** (section 2.2) that specifies a value that uniquely identifies the record on which a **FixupRow** operation (section 3.1.5.1.6) is to be performed. MUST be present. MUST NOT be **json-null**.

SupportingFieldIndexes: A **JsonArrayOfInt** (section 2.2), that specifies the indexes of the fields whose values are used for performing **FixupRow** operation. MUST be present. MUST have at least one element.

SupportingFieldValues: A **JsonArrayOfString** (section 2.2) that specifies the values of **SupportingFieldIndexes**. MUST be present. MUST have at least one element.

2.2.1.6 FormatInfo

The **FormatInfo** complex type specifies the format for a field in a **Source** (section 3.1.1). The format is used by the protocol server to verify the value of the field sent by the protocol client and also to convert the value of field to its format before returning the value to the protocol client.

The **FormatInfo** complex type is an input as well as output used by some protocol operations.

The following ABNF references types specified by section 2.2 and applies to the **FormatInfo**, unless extended or overridden by an operation specification.

```

FormatInfo = json-object
Currency = json-quotation-mark "Currency" json-quotation-mark json-name-separator
json-string
Format = json-quotation-mark "Format" json-quotation-mark json-name-separator json-
string
Precision = json-quotation-mark "Precision" json-quotation-mark json-name-separator
json-int

```

FormatInfo MUST contain three **json-member** instances, and each instance MUST be different. Each instance MUST be one of the following: **Currency**, **Format**, and **Precision**.

Currency: A **json-string** (section 2.2) that specifies the currency symbol for the value of a field of a **Source** (section 3.1.1) in the database application. MUST be set to **json-null** (section 2.2) when the **Format** of the value of the field is not "currency".

Format: A **json-string** (section 2.2) that specifies the format for the value of a field of a **Source** (section 3.1.1) in the database application. MUST either be **json-null** (section 2.2) or MUST have one of the following values.

Value	Meaning
"general date"	The field is formatted as general date. For example, 01/02/2011 01:20:30 PM.

Value	Meaning
"long date"	The field is formatted as long date. For example, Monday, Jan 01, 2011.
"short date"	The field is formatted as short date. For example, 01/01/11.
"long time"	The field is formatted as long time. For example, 5:34:45 PM.
"medium time"	The field is formatted as medium time. For example, 5:34.
"short time"	The field is formatted as short time. For example, 17.
"general number"	The number format is general. For example, 123,456.
"standard"	The number format applied is standard. For example, 1234,567.89.
"fixed"	The number format applied is fixed. For example, 1234.56.
"scientific"	The number format applied is scientific. For example, 123E+03.
"percent"	The number format applied is percentage. For example, 12.3%.
"currency"	The number format applied is currency. For example, \$12.34.

Precision: A **json-int** (section 2.2) that specifies the maximum number of digits present after the decimal point in the value of a field of a **Source** (section 3.1.1) in the database application. **MUST** be a number between 0 and 15 and **MUST** only be present when **Format** is one of the following:

- "general number"
- "standard"
- "fixed"
- "scientific"
- "percent"
- "currency"

2.2.1.7 PagingInfo

The **PagingInfo** complex type specifies which **Page** (section 3.1.1) of data from the cached result set or from the **Source** (section 3.1.1) is requested by the protocol client.

The **PagingInfo** complex type is used as an input as well as output by the some protocol operations.

The following ABNF references types specified by section 2.2, and applies to the **PagingInfo** complex type unless extended or overridden by an operation specification.

```
PagingInfo = json-object
FirstRow = json-quotation-mark "FirstRow" json-quotation-mark json-name-separator
           json-int
PageSize = json-quotation-mark "PageSize" json-quotation-mark json-name-separator
           json-int
```

```

Moniker = json-quotation-mark "Moniker" json-quotation-mark json-name-separator
(json-string / json-null)
UseCache = json-quotation-mark "UseCache" json-quotation-mark json-name-separator
json-bool
CacheCommands = json-quotation-mark "CacheCommands" json-quotation-mark json-name-
separator CacheCommands
SortExpression = json-quotation-mark "SortExpression" json-quotation-mark json-
name-separator json-string
Filter = json-quotation-mark "Filter" json-quotation-mark json-name-separator
FilterInfo
TotalRows = json-quotation-mark "TotalRows" json-quotation-mark json-name-separator
json-int
SessionId = json-quotation-mark "SessionId" json-quotation-mark json-name-separator
json-string
RetrieveExactRowCount = json-quotation-mark "RetrieveExactRowCount" json-quotation-
mark json-name-separator json-bool
RowKey = json-quotation-mark "RowKey" json-quotation-mark json-name-separator json-
int

```

PagingInfo MUST contain at least one and no more than nine **json-member** instances, and each instance MUST be different. Each instance MUST be one of the following: **CacheCommands**, **Filter**, **FirstRow**, **Moniker**, **PageSize**, **SessionId**, **SortExpression**, **TotalRows**, or **UseCache**.

CacheCommands: A **CacheCommands** (section [2.2.2.1](#)) that specifies the operations to perform on the result set.

Filter: A **FilterInfo** (section [2.2.1.4](#)) that specifies how to filter the result set.

FirstRow: A **json-int** (section 2.2) that specifies the index of the first record to return from the **Source** (section 3.1.1) specified by the **SelectCommand** element of the **SharedDataBaseInfo** (section [2.2.1.12](#)).

- MUST be present.
- MUST be non-negative.
- MUST be less than total records in **Source**.

Moniker: A **json-string** (section 2.2) that specifies the identifier for a result set that was specified by the protocol client. If a **Moniker** is specified in the **session** then it MUST be used as the **moniker** element for the input to all **GetData** service operations (section [3.1.5.1.1](#)) within the same session.

PageSize: A **json-int** (section 2.2) that specifies the maximum number of records to return. MUST be non-negative.

SessionId: A **SessionIdentifier**, as specified by [\[MS-ADR\]](#) section 3.1.1.2, that specifies a unique identifier for the current session that was previously received from the protocol server in the response of the first **GetData** service operation (section 3.1.5.1.1).

SortExpression: A **json-string** (section 2.2) that specifies the sort order to apply to the result set. If the **CacheCommands** contains **ApplySort**:

- MUST be present.
- MUST contain a **CT_AdhocOrdering**, as specified in [\[MS-AXL2\]](#) section 2.2.3.5.

TotalRows: A **json-int** (section 2.2) that specifies the total number of records in the **Source**. This element is populated by the protocol server, and sent as a part of **PagingInfo**, in the response body of all protocol operations. MUST be present when included in a paging response.

UseCache: MUST be ignored.

RetrieveExactRowCount: A **json-bool** (section 2.2) that specifies whether the protocol server retrieves the exact number of records or an approximation of the number of records in the **Source**. The protocol server retrieves the exact number of records if the value of **RetrieveExactRowCount** is **json-true**. If this element is not specified, a default value of **json-false** is used.

RowKey: A **json-int** (section 2.2) that specifies the value of the **primary key** field of the record to return from the **Source** (section 3.1.1) specified by the **SelectCommand** element of the **SharedDataBaseInfo** (section 2.2.1.12).

- MUST be non-negative.

2.2.1.8 ParameterValue

The **ParameterValue** complex type is an input used by all protocol operations.

The following applies to the **ParameterValue** complex type, unless extended or overridden by an operation specification.

The **ParameterValue** type serves as a container for the name of a **CT_Parameter** ([\[MS-AXL2\]](#) section 2.2.3.12) and its associated value. Some **Sources** (section [3.1.1](#)) in a database application are parameterized, and need parameter values to open them. Hence, this complex type is sent along with **SelectCommand** in a **SharedDataBaseInfo** (section [2.2.1.12](#)) in the request body of all protocol operations. The following ABNF references types specified by section [2.2](#).

```
ParameterValue = json-object
Name = json-quotation-mark "Name" json-quotation-mark json-name-separator json-string
Value = json-quotation-mark "Value" json-quotation-mark json-name-separator json-value
```

ParameterValue MUST contain two **json-member** instances that are not **json-null**, and each instance MUST be different. Each instance MUST be one of the following: **Name** and **Value**.

Name: A **json-string** (section 2.2) that specifies a value that uniquely identifies the **ParameterValue** within a collection.

Value: A **json-value** (section 2.2) that specifies data associated with the given **Name**.

2.2.1.9 RecordSet

The **RecordSet** is a complex type that is output by all the protocol operations. The **RecordSet** is specified by the **Result** element of the **ServiceResult** complex type (section [2.2.1.11](#)).

The following ABNF references types specified by section [2.2](#) and applies to the **RecordSet**, unless extended or overridden by an operation specification.

```
RecordSet = json-object
Values = json-quotation-mark "Values" json-quotation-mark json-name-separator
JsonArrayOfArrayOfAnyType
Localized = json-quotation-mark "Localized" json-quotation-mark json-name-separator
JsonArrayOfArrayOfString
Paging = json-quotation-mark "Paging" json-quotation-mark json-name-separator
PagingInfo
Fields = json-quotation-mark "Fields" json-quotation-mark json-name-separator json-begin-array [FieldSchema *( json-value-separator FieldSchema) ] json-end-array
```

```

FormatInfos = json-quotation-mark "FormatInfos" json-quotation-mark json-name-
separator json-begin-array [FormatInfo *( json-value-separator FormatInfo) ] json-
end-array
AutoSumValues = json-quotation-mark "AutoSumValues" json-quotation-mark json-name-
separator jsonArrayOfAnyType
CurrentUserPermissions = json-quotation-mark "CurrentUserPermissions" json-
quotation-mark json-name-separator CurrentUserPermissions
RelatedFieldInfos = json-quotation-mark "RelatedFieldInfos" json-quotation-mark
json-name-separator json-begin-array [FieldSchema *( json-value-separator
FieldSchema) ] json-end-array

```

RecordSet MUST contain at least one **json-member** instance from the following **json-member** instances. All **json-member** instances MUST be unique.

Values: A **JsonArrayOfArrayOfAnyType** (section 2.2) that specifies a record array. Each record is an array of values for the fields (3). Each field value at an ordinal in a record array is the value of the **FieldSchema** (section 2.2.1.3) at the same ordinal in the **Fields** array **RecordSet**, returned in the **Result** element of the **ServiceResult** element from a previous call to the **GetData** operation (section 3.1.5.1.1). MUST be present.

Localized: A **JsonArrayOfArrayOfArrayOfString** (section 2.2) that specifies the formatted value for each value in the **Values** array at the same ordinal in the **FormatInfos** element in **SharedDataBaseInfo** (section 2.2.1.12). MUST be present.

Paging: A **PagingInfo**, as specified by section 2.2.1.7. MUST be present.

Fields: A **json-array** of **FieldSchema** ([MS-ADR] section 2.2.4.19) that specifies the schema of the fields of the **Source** (section 3.1.1), which is specified by the **SelectCommand** of the **SharedDataBaseInfo** complex type (section 2.2.1.12). MUST be present.

FormatInfos: A **json-array** of **FormatInfo** (section 2.2.1.6) that specifies the format information for the fields in the **Source**. MUST be present.

AutoSumValues: A **JsonArrayOfAnyType** (section 2.2) that specifies the results of aggregate functions, which are specified by the **AutoSumFunctions** of the **SharedDataBaseInfo** complex type (section 2.2.1.12), when applied to the fields specified by the **AutoSumFields** of the **SharedDatabaseInfo**.

CurrentUserPermissions: A **CurrentUserPermissions** (section 2.2.1.2) that specifies the permissions the user has for the database application.

RelatedFieldInfos: A **json-array** of **FieldSchema** ([MS-ADR] section 3.1.4.7.3.1) that specifies the schema of the related fields of the **Source** (section 3.1.1), which is specified by the **SelectCommand** of the **SharedDataBaseInfo** complex type (section 2.2.1.12).

2.2.1.10 ServiceError

The **ServiceError** complex type is an output used by all protocol operations. It is used to return alert (2) information to the protocol client.

The following ABNF references types specified by section 2.2 and applies to the **ServiceError**, unless extended or overridden by an operation specification.

```

ServiceError = json-object
Message = json-quotation-mark "Message" json-quotation-mark json-name-separator
ClientMessage
Caption = json-quotation-mark "Caption" json-quotation-mark json-name-separator
(json-string / json-null)

```

```

HelpText = json-quotation-mark "HelpText" json-quotation-mark json-name-separator
(json-string / json-null)
HelpId = json-quotation-mark "HelpId" json-quotation-mark json-name-separator
(json-string / json-null)
Severity = json-quotation-mark "Severity" json-quotation-mark json-name-separator
MessageSeverity
Number = json-quotation-mark "Number" json-quotation-mark json-name-separator json-
value

```

ServiceError MUST contain at least five and no more than six **json-member** instances, and each instance MUST be different. Each instance MUST be one of the following: **Caption**, **HelpId**, **HelpText**, **Message**, **Number**, and **Severity**.

Caption: A **json-string** (section 2.2) that specifies the **caption** for the **alert**. MUST be present.

HelpId: A **json-string** (section 2.2) that specifies the **Help topic identifier**.

HelpText: A **json-string** (section 2.2) that specifies a description of the help topic. MUST be present.

Message: A **ClientMessage** (section [2.2.1.1](#)) that specifies a description of the alert. MUST be present. MUST NOT be empty.

Number: A **json-value** (section 2.2) that specifies an error code that is associated with this alert. MUST either be **json-null** or **json-int**.

Severity: A **MessageSeverity** (section [2.2.2.3](#)) that specifies the severity of the alert. MUST be present.

2.2.1.11 ServiceResult

The **ServiceResult** complex type is an output used by all protocol operations.

The following ABNF references types specified by section [2.2](#) and applies to the **ServiceResult**, unless extended or overridden by an operation specification.

```

ServiceResult = json-object
Result = json-quotation-mark "Result" json-quotation-mark json-name-separator json-
value
Error = json-quotation-mark "Error" json-quotation-mark json-name-separator
ServiceError

```

ServiceResult MUST contain at least one and no more than two **json-member** instances, and each instance MUST be different. Each instance MUST be one of the following: **Error**, and **Result**.

Error: A **ServiceError**, as specified in section [2.2.1.10](#).

Result: A **json-value** (section 2.2) that specifies the result of the protocol operation.

After the protocol server has finished the execution of the protocol operation, it creates the **ServiceResult** object, and then sends it in the Response body of the protocol operation. The **Result** element of the **ServiceResult** contains the result of the protocol operation, if it finished successfully, otherwise the **Error** element will contain the error information.

2.2.1.12 SharedDataBaseInfo

The **SharedDataBaseInfo** complex type specifies the **Source**, the fields from this **Source**, the sort order and restriction criteria of the records that are retrieved from database application.

The **SharedDataBaseInfo** complex type is an input used by all protocol operations.

The following ABNF references types specified by section 2.2, and applies to the **SharedDataBaseInfo** unless extended or overridden by an operation specification.

```
SharedDataBaseInfo = json-object
SessionId = json-quotation-mark "SessionId" json-quotation-mark json-name-separator
(json-string | json-null)
SelectCommand = json-quotation-mark "SelectCommand" json-quotation-mark json-name-
separator (json-string | json-null)
ParameterValues = json-quotation-mark "ParameterValues" json-quotation-mark json-name-
separator json-begin-array [ParameterValue *( json-value-separator ParameterValue) ]
json-end-array
OriginalCommand = json-quotation-mark "OriginalCommand" json-quotation-mark json-name-
separator (json-string | json-null)
AggregateExpressions = json-quotation-mark "AggregateExpressions" json-quotation-mark
json-name-separator (json-string | json-null)
AggregateFormatInfos = json-quotation-mark "AggregateFormatInfos" json-quotation-mark
json-name-separator json-begin-array [FormatInfo *( json-value-separator FormatInfo)
] json-end-array
Restriction = json-quotation-mark "Restriction" json-quotation-mark json-name-
separator (json-string | json-null)
Ordering = json-quotation-mark "Ordering" json-quotation-mark json-name-separator
(json-string | json-null)
AllowEdits = json-quotation-mark "AllowEdits" json-quotation-mark json-name-separator
(json-string | json-null)
AllowAdditions = json-quotation-mark "AllowAdditions" json-quotation-mark json-name-
separator (json-string | json-null)
AllowDeletions = json-quotation-mark "AllowDeletions" json-quotation-mark json-name-
separator (json-string | json-null)
DataEntry = json-quotation-mark "DataEntry" json-quotation-mark json-name-separator
json-bool
FieldNames = json-quotation-mark "FieldNames" json-quotation-mark json-name-separator
(json-string | json-null)
FormatInfos = json-quotation-mark "FormatInfos" json-quotation-mark json-name-
separator json-begin-array [FormatInfo *( json-value-separator FormatInfo) ] json-
end-array
DataLevelFormat = json-quotation-mark "DataLevelFormat" json-quotation-mark json-name-
separator json-bool
ShowHeaders = json-quotation-mark "ShowHeaders" json-quotation-mark json-name-
separator json-bool
InitialPage = json-quotation-mark "InitialPage" json-quotation-mark json-name-
separator (json-string | json-null)
FetchSchema = json-quotation-mark "FetchSchema" json-quotation-mark json-name-
separator json-bool
FetchKeyFields = json-quotation-mark "FetchKeyFields" json-quotation-mark json-name-
separator json-bool
FetchDisplayFields = json-quotation-mark "FetchDisplayFields" json-quotation-mark json-
name-separator json-bool
DoNotPrefetchImages = json-quotation-mark "DoNotPrefetchImages" json-quotation-mark
json-name-separator json-bool
AutoSumFields = json-quotation-mark "AutoSumFields" json-quotation-mark json-name-
separator JsonArrayOfString
AutoSumFunctions = json-quotation-mark "AutoSumFunctions" json-quotation-mark json-
name-separator JsonArrayOfString
```

SharedDataBaseInfo MUST contain at least one **json-member** instance from the following **json-member** instances. All **json-member** instances MUST be unique.

AggregateExpressions: MUST NOT be used.

AggregateFormatInfos: MUST NOT be used.

AllowAdditions: MUST NOT be used.

AllowDeletions: MUST NOT be used.

AllowEdits: MUST NOT be used.

DataEntry: MUST NOT be used.

DataLevelFormat: MUST NOT be used.

FetchSchema: A **json-bool** (section 2.2) that specifies whether the protocol server fetches the schema information of the **SelectCommand** element. The protocol server fetches the schema information if the value of **FetchSchema** is **json-true**; otherwise, the protocol server does not fetch schema information. If this element is not specified, a default value of **json-true** is used.

FieldNames: The fields from the **SelectCommand** for which data is to be fetched.

FormatInfos: A **json-array** of **FormatInfo** instances, as specified in section [2.2.1.6](#).

InitialPage: MUST NOT be used.

Ordering: A **json-string** (section 2.2) that specifies the sort order. If present, the protocol server applies this sort order to the data and returns the sorted data. This element MUST contain a **CT_AdHocOrdering** complex type, as specified in [\[MS-AXL2\]](#) section 2.2.3.5.

OriginalCommand: MUST NOT be used.

ParameterValues: The value of each parameter, as specified in [\[MS-AXL2\]](#) section 2.2.3.17, required in the **SelectCommand** element. This element MUST be present for every parameter defined in **SelectCommand**.

- The **Key** element is the name of the parameter.
- The **Value** element is the value of the parameter in the correct data type.

Restriction: A **CT_Expression** complex type, as specified in [\[MS-AXL2\]](#) section 2.2.3.1. Only records in **SelectCommand** that return "true" for this restriction expression are included in the **Result Set** (section [3.1.1](#)).

SelectCommand: A **json-string** (section 2.2) that specifies the **Source** (section 3.1.1) from which data is to be retrieved.

SessionId: A **json-string** (section 2.2) that specifies a unique identifier for the session as specified in [\[MS-ADR\]](#) section 3.1.1.2.

ShowHeaders: MUST NOT be used.

FetchKeyFields: A **json-bool** (section 2.2) that specifies whether the protocol server fetches the values of the primary key fields of the **SelectCommand** element. The protocol server fetches the values of the primary key fields if the value of **FetchKeyFields** is **json-true**; otherwise, the protocol server does not fetch the values of the primary key fields. If this element is not specified, a default value of **json-false** is used.

FetchDisplayFields: A **json-bool** (section 2.2) that specifies whether the protocol server fetches the display values of the **foreign key** fields of the **SelectCommand** element. The protocol server fetches the display values of the **foreign key** fields if the value of **FetchDisplayFields** is **json-true**;

otherwise, the protocol server does not fetch the display values of the **foreign key** fields. If this element is not specified, a default value of **json-false** is used

DoNotPrefetchImages: A **json-bool** (section 2.2) that specifies whether the protocol server fetches and caches the values of the image fields of the **SelectCommand** element. The protocol server does not fetch and cache the values of the image fields if the value of **DoNotPrefetchImages** is **json-true**; otherwise, the protocol server fetches and caches the values of the image fields. If this element is not specified, a default value of **json-false** is used.

AutoSumFields: A **JsonArrayOfString** (section 2.2) that specifies the names of fields for which the protocol server applies the **AutoSumFunctions**.

AutoSumFunctions: A **JsonArrayOfString** (section 2.2) that specifies the aggregate functions to be applied to the **AutoSumFields**. The number of elements in **AutoSumFunctions** MUST be exactly equal to the number of elements in the **AutoSumFields**.

2.2.1.13 UpdateRecord

The **UpdateRecord** complex type specifies the new values and original values for fields in an **Updatable Source** (section 3.1.1) that are used for insert, update, and delete operations.

The **UpdateRecord** type is an input used by some protocol operations.

The following ABNF references types specified by section 2.2 and applies to **UpdateRecord**, unless extended or overridden by an operation specification.

```
UpdateRecord = json-object
OriginalValues = json-quotation-mark "OriginalValues" json-quotation-mark json-name-
separator JsonArrayOfArrayOfAnyType
NewValues = json-quotation-mark "NewValues" json-quotation-mark json-name-separator
JsonArrayOfArrayOfString
Paging = json-quotation-mark "Paging" json-quotation-mark json-name-separator
PagingInfo
ReturnDataMacroIds = json-quotation-mark "ReturnDataMacroIds" json-quotation-mark json-
name-separator json-bool
```

UpdateRecord MUST contain at least one and no more than four **json-member** instances, and each instance MUST be different. Each instance MUST be one of the following: **NewValues**, **OriginalValues**, **Paging**, and **ReturnDataMacroIds**.

NewValues: A **JsonArrayOfArrayOfString** (section 2.2) that specifies the new values of a collection of records which need to be inserted, updated, or deleted. The protocol server will insert, update, or delete the records in a **Source** (section 3.1.1) with the values that are provided by this element.

OriginalValues: A **JsonArrayOfArrayOfAnyType** (section 2.2) that specifies the existing values of a collection of records in a **Source** (section 3.1.1), which need to be inserted, updated, or deleted.

Paging: A **PagingInfo** complex type, as specified in section 2.2.1.7. MUST be present.

ReturnDataMacroIds: MUST NOT be used.

2.2.2 Simple Types

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

Simple Type	Description
CacheCommands	The CacheCommands simple type as specified in [MS-ADR] section 2.2.5.5 . See section 2.2.2.1 for more details.
ClientMessageID	A ClientMessageID simple type, as specified in [MS-ADR] section 2.2.5.6 . See section 2.2.2.2 for more details.
MessageSeverity	Specifies the severity of a ServiceError (section 2.2.1.10) sent from protocol server to protocol client. See section 2.2.2.3 for more details.

2.2.2.1 CacheCommands

The meaning of **CacheCommands** is specified by [\[MS-ADR\] section 2.2.5.5](#). The format of the **CacheCommands** simple type is specified by this section. The following ABNF references types specified by section [2.2](#).

```
CacheCommands = json-int;
```

The value of **CacheCommands** MUST be one of the values in the following table.

Value	Meaning
0x01	RefreshData, See [MS-ADR] section 2.2.5 .
0x02	ApplyFilter, See [MS-ADR] section 2.2.5 .
0x04	ClearFilter, See [MS-ADR] section 2.2.5 .
0x08	ApplySort, See [MS-ADR] section 2.2.5 .
0x10	RetrieveImage, See [MS-ADR] section 2.2.5 .

2.2.2.2 ClientMessageID

The meaning of **ClientMessageID** is specified by [\[MS-ADR\] section 2.2.5.6](#). The format of the **ClientMessageID** simple type is specified by this section. The following ABNF references types specified by section [2.2](#).

```
ClientMessageID = json-string;
```

The value of the **ClientMessageID** string MUST be one of the values in the following table.

Value	Meaning
None	See [MS-ADR] section 2.2.5.6
PassthroughMessage	See [MS-ADR] section 2.2.5.6
WebServiceUnavailableMessage	See [MS-ADR] section 2.2.5.6
WebServiceCanceledMessage	See [MS-ADR] section 2.2.5.6

Value	Meaning
DataMacroStopWaiting	See [MS-ADR] section 2.2.5.6
GenericError	See [MS-ADR] section 2.2.5.6
NotifyRecordUpdated	See [MS-ADR] section 2.2.5.6
NotifyRecordsDeleted	See [MS-ADR] section 2.2.5.6
NotifyCannotDelete	See [MS-ADR] section 2.2.5.6
NotifyCannotSave	See [MS-ADR] section 2.2.5.6
NotifyCannotSaveDescription	See [MS-ADR] section 2.2.5.6
SubFormNestingLimitExceededError	See [MS-ADR] section 2.2.5.6
DivisionByZero	See [MS-ADR] section 2.2.5.6
TypeMismatch	See [MS-ADR] section 2.2.5.6
NoCurrentRecord	See [MS-ADR] section 2.2.5.6
RecordNotUpdatable	See [MS-ADR] section 2.2.5.6
WhereConditionTooLong	See [MS-ADR] section 2.2.5.6
InvalidArgument	See [MS-ADR] section 2.2.5.6
InvalidContainerControlName	See [MS-ADR] section 2.2.5.6
ActionArgumentNotFound	See [MS-ADR] section 2.2.5.6
FormCannotClose	See [MS-ADR] section 2.2.5.6
AttachmentMustSave	See [MS-ADR] section 2.2.5.6
InvalidExpression	See [MS-ADR] section 2.2.5.6
AttachmentUploadSucceed	See [MS-ADR] section 2.2.5.6
AttachmentDeleted	See [MS-ADR] section 2.2.5.6
FormOrReportNotBound	See [MS-ADR] section 2.2.5.6
InvalidPropertyOrMethod	See [MS-ADR] section 2.2.5.6
InvalidUrl	See [MS-ADR] section 2.2.5.6
UserInterfaceMacroLoadFailure	See [MS-ADR] section 2.2.5.6
CannotFindMacroObject	See [MS-ADR] section 2.2.5.6
CannotGoToSpecifiedRecord	See [MS-ADR] section 2.2.5.6
InvalidConditionExpression	See [MS-ADR] section 2.2.5.6
CannotGoToSubformOrTabPage	See [MS-ADR] section 2.2.5.6
InvalidControl	See [MS-ADR] section 2.2.5.6
CannotFocusOnControl	See [MS-ADR] section 2.2.5.6
InvalidControlValue	See [MS-ADR] section 2.2.5.6

Value	Meaning
NoControlName	See [MS-ADR] section 2.2.5.6
CannotFindErrorHandlingMacro	See [MS-ADR] section 2.2.5.6
Invalid SetPropertyControl	See [MS-ADR] section 2.2.5.6
LocalVarNameNotSpecified	See [MS-ADR] section 2.2.5.6
LocalVarExpressionNotSpecified	See [MS-ADR] section 2.2.5.6
LocalVariableNameInvalid	See [MS-ADR] section 2.2.5.6
TempVarNameNotSpecified	See [MS-ADR] section 2.2.5.6
TempVariableNameInvalid	See [MS-ADR] section 2.2.5.6
CommandNotAvailable	See [MS-ADR] section 2.2.5.6
ExpressionErrorCannotSet	See [MS-ADR] section 2.2.5.6
TooManyRunMacros	See [MS-ADR] section 2.2.5.6
InvalidBrowseToPathArgument	See [MS-ADR] section 2.2.5.6
InvalidObjectReference	See [MS-ADR] section 2.2.5.6
FormNotFoundRep	See [MS-ADR] section 2.2.5.6
ControlDoesNotSupportRequery	See [MS-ADR] section 2.2.5.6
ReadOnlyNote	See [MS-ADR] section 2.2.5.6
TooMuchParameterData	See [MS-ADR] section 2.2.5.6
OperationNotSupportedOnSubReports	See [MS-ADR] section 2.2.5.6
ExpressionErrorInfo	See [MS-ADR] section 2.2.5.6
InvalidPropertyValue	See [MS-ADR] section 2.2.5.6
AutocompleteWaitingForValidationMessage	See [MS-ADR] section 2.2.5.6
AutocompleteInvalidDataMessage	See [MS-ADR] section 2.2.5.6
TooManyTermsInFilter	See [MS-ADR] section 2.2.5.6
InvalidValueForFieldMessage	See [MS-ADR] section 2.2.5.6
InvalidValueForControlMessage	See [MS-ADR] section 2.2.5.6
ImageUploadFailureNotificationMessage	See [MS-ADR] section 2.2.5.6
NoResultsFromFilter	See [MS-ADR] section 2.2.5.6

2.2.2.3 MessageSeverity

The **MessageSeverity** simple type is used to specify the severity of a **ServiceError** (section [2.2.1.10](#)) sent to the protocol client. The following ABNF references types specified by section [2.2.](#)

MessageSeverity = json-string;

The value of the **MessageSeverity** string MUST be one of the values in the following table.

Value	Meaning
Info	The ServiceError indicates that an informational message has been sent from the protocol server.
Warn	The ServiceError indicates that a warning has been sent from the protocol server.
Error	The ServiceError indicates that an error has occurred.

3 Protocol Details

3.1 Server Details

The protocol server receives request messages from the protocol client in JSON format. The request messages can be either to retrieve, insert, update or delete data from the database application. The protocol server processes the request, and then returns the result set and alert in the **Result** and **Error** elements of **ServiceResult** (section [2.2.1.11](#)) object to the protocol client. The protocol server never initiates communication with other endpoints of the protocol.

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.

This protocol includes the following Abstract Data Model elements, which are directly accessed from Access Services Data Run Time protocol as specified in [\[MS-ADR\]](#) section 3.1.1:

- **Session**
- **Request**
- **Source**
- **Updatable Source**

The following elements are specific to this protocol:

- **Page:** A partial result set, which is part of a sequence of partial result sets of the same size.

3.1.2 Timers

No new timers are required beyond those specified in [\[MS-ADR\]](#) section 3.1.2.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

None.

3.1.5 Message Processing Events and Sequencing Rules

The following table summarizes the operations in this protocol.

Resources	Description
AccessPortal	Specifies a resource which operates against a site that contains the database application. See section 3.1.5.1 for more details.

The responses to all the operations can result in the following status codes.

Status code	Description
200	This status code indicates that operation finished successfully.

3.1.5.1 AccessPortal

This resource operates against a site that is identified by a **URL** that is known by protocol clients. The protocol server endpoint is formed by appending `"/_vti_bin/acccsvc/accessportal.json"` to the URL of the site, for example: `http://www.example.com/Repository/_vti_bin/acccsvc/accessportal.json`.

The operation path is obtained by appending the operation name to the endpoint, for example, `"http://www.example.com/Repository/_vti_bin/acccsvc/accessportal.json/<serviceName>"`

The URL parameters are defined by the following ABNF syntax:

```
serviceName = STRING
```

serviceName: The unique name of the hosted service operation.

Operation	Description
GetData	Retrieves a subset of rows from a Source (section 3.1.1) in a database application. See section 3.1.5.1.1 for more details.
InsertRecords	Inserts records into an Updatable Source (section 3.1.1) in a database application. See section 3.1.5.1.2 for more details.
UpdateRecords	Updates records into an Updatable Source (section 3.1.1) in a database application. See section 3.1.5.1.3 for more details.
DeleteRecords	Deletes records from an Updatable Source (section 3.1.1) in a database application. See section 3.1.5.1.4 for more details.
GetDistinctValues	Retrieves distinct values for a field from a Source (section 3.1.1) in a database application. See section 3.1.5.1.5 for more details.
FixupRow	Performs modeling of a hypothetical update for an Updatable Source (section 3.1.1) in a database application. See section 3.1.5.1.6 for more details.
GetSearchData	Retrieves records from Source (section 3.1.1). See section 3.1.5.1.7 for more details.

3.1.5.1.1 GetData

This operation fetches a subset of rows from a **Source** (section [3.1.1](#)), which is passed in as **SelectCommand** element of **SharedDataBaseInfo** (section [2.2.1.12](#)).

This operation is transported by an HTTP **POST**.

The **serviceName** as specified in section [3.1.5.1](#) MUST be "GetData".

The operation can be invoked through the following **URI**:

http://www.example.com/Repository/_vti_bin/acccsvc/accessportal.json/GetData.

During this operation, the protocol server receives a JSON request that contains the **SharedDataBaseInfo** (section 2.2.1.12) and **PagingInfo** (section 2.2.1.7) objects. The protocol server then processes the request, and then responds with a JSON response that contains **ServiceResult** (section 2.2.1.11). The **Values** element of **RecordSet** (section 2.2.1.9), which is returned as **Result** element of the **ServiceResult** contains the rows of data which protocol client had requested.

In the event of an application error on the protocol server during this operation, the **ServiceError**, as specified in section 2.2.1.10, MUST be present in **ServiceResult** (section 2.2.1.11).

The processing details for this operation are specified by section 3.1.5.1.1.3.

3.1.5.1.1.1 Request Body

The **SharedDataBaseInfo** (section 2.2.1.12) and **PagingInfo** (section 2.2.1.7) objects in Request Body determine which records are fetched from a **Source** (section 3.1.1), which is specified by the **SelectCommand** element of the **SharedDataBaseInfo**. The following ABNF references types specified by section 2.2.

```
GetDataRequest = dataBaseInfo-member pagingInfo-member
dataBaseInfo-member = json-quotation-mark "dataBaseInfo" json-quotation-mark json-name-separator SharedDataBaseInfo
pagingInfo-member = json-quotation-mark "pagingInfo" json-quotation-mark json-name-separator PagingInfo
```

dataBaseInfo: A **SharedDataBaseInfo** (section 2.2.1.12) that specifies the **Source**, the fields from this **Source**, the sort order and restriction criteria of the records that are retrieved from the database application.

pagingInfo: A **PagingInfo** (section 2.2.1.7) that specifies which records are retrieved from the database application. MUST be present.

3.1.5.1.1.2 Response Body

The records from the **Source** (section 3.1.1) which is specified by the **SelectCommand** element of **SharedDataBaseInfo** (section 2.2.1.12) are returned to the protocol client in the **Values** element of **RecordSet** (section 2.2.1.9), which is returned as the **Result** element of the **ServiceResult** (section 2.2.1.11) in the Response Body. If there is any error on the protocol server, a **ServiceError** (section 2.2.1.10) is returned in the **Error** element of **ServiceResult**. The following ABNF references types specified by section 2.2.

```
GetDataResult = json-null | serviceResult-member
serviceResult-member = json-quotation-mark "d" json-quotation-mark json-name-separator ServiceResult
```

GetDataResult: A **ServiceResult** (section 2.2.1.11) that specifies the records returned by protocol server to the protocol client. MUST be present.

3.1.5.1.1.3 Processing Details

The protocol client sends a request message, and the protocol server responds as specified in [MS-ADR] section 3.1.4.4, except as follows:

- The protocol server retrieves data from the given **Source** (section [3.1.1](#)) which is supplied as the **SelectCommand** element of the **SharedDataBaseInfo** (section [2.2.1.12](#)), which is an input element of the Request Body (section [3.1.5.1.1.1](#)).
- Only the fields that are specified by **FieldNames** input element of **SharedDataBaseInfo** are retrieved.
- If the **FetchSchema** element of **SharedDataBaseInfo** is true, the protocol server also returns the schema information about the fields in **Source**, which are specified by **Fields** element of **SharedDataBaseInfo**. This schema information is returned via the **Fields** element of the **RecordSet** (section [2.2.1.9](#)).
- If the request sent to the protocol server is the first GetData request in the session, then the protocol server MUST return the **SessionIdentifier** ([MS-ADR] section 3.1.1.2) in the **SessionId** element of the **PagingInfo** (section [2.2.1.7](#)). The **PagingInfo** type is returned as an element in the **RecordSet**, which is returned as the **Result** element of the **ServiceResult** (section [2.2.1.11](#)) complex type.

In the event of an application error on the protocol server during this operation, the **ServiceError**, as specified in section [2.2.1.10](#), MUST be present in **ServiceResult** (section 2.2.1.11).

3.1.5.1.2 InsertRecords

This operation inserts records into an **Updatable Source** (section [3.1.1](#)) specified in the **SelectCommand** element of the **SharedDataBaseInfo** element in the request body (section [2.2.1.12](#)).

The protocol server receives a JSON request that contains the **SharedDataBaseInfo** and **UpdateRecord** (section [2.2.1.13](#)) objects. The protocol server processes the request and responds with a JSON response that contains the **ServiceResult** (section [2.2.1.11](#)). The **Values** element of **RecordSet** (section [2.2.1.9](#)) returned as the **Result** element of the **ServiceResult** contains the records of data which the protocol server inserted.

In the event of an application error on the protocol server, the **ServiceError** element (section [2.2.1.10](#)) MUST be present in **ServiceResult** (section 2.2.1.11).

This operation is transported by an HTTP **POST**.

The **serviceName** as specified in section [3.1.5.1](#) MUST be "InsertRecords".

The processing details for this operation are specified by section [3.1.5.1.2.3](#).

3.1.5.1.2.1 Request Body

The **SharedDataBaseInfo** (section [2.2.1.12](#)) and **UpdateRecord** (section [2.2.1.13](#)) objects in Request Body determine the values for the fields of the records to be inserted into the **Updatable Source** (section [3.1.1](#)). The **Updatable Source** is specified by the **FieldSchema** elements in **RecordSet** (section [2.2.1.9](#)) returned as the **Result** element in the Response Body of a prior **GetData** operation (section [3.1.5.1.1.2](#)).

The values to be inserted are specified by the **NewValues** element of **UpdateRecord**. These values MUST be formatted according to the **FormatInfo** element of **SharedDataBaseInfo**. The **SessionId** element of **SharedDataBaseInfo** MUST be the same as the **SessionId** element of the **PagingInfo** which is an element of **RecordSet** (section 2.2.1.9) returned as the **Result** element in Response Body of the first **GetData** operation.

```
InsertRecordsRequest = dataBaseInfo-member updateRecord-member
dataBaseInfo-member = json-quotation-mark "dataBaseInfo" json-quotation-mark json-name-
separator SharedDataBaseInfo
```

```
updateRecord-member = json-quotation-mark "updateRecord" json-quotation-mark json-name-
separator UpdateRecord
```

dataBaseInfo: A **SharedDataBaseInfo** (section 2.2.1.12) that specifies information about the **Updatable Source** (section 3.1.1) into which records are to be inserted.

updateRecord: An **UpdateRecord** (section 2.2.1.13) that specifies the values for the fields of the records to be inserted.

- The number of fields in each record MUST be the same for all the records, and MUST match the number and ordering of fields in the **FieldNames** element of **dataBaseInfo**.
- The value of primary key fields in **NewValues** element MUST be "null".
- Each field value at an ordinal in a record array from the **NewValues** element MUST be formatted according to the **FormatInfo** at the same ordinal in the **FormatInfos** array of the **dataBaseInfo**.
- The **OriginalValues** element of the **UpdateRecord** MUST be ignored.
- The field value at an ordinal in a record array from the **NewValues** element MUST be "null" if the **ReadOnly** element of the **FieldSchema** element at the same ordinal in the **Fields** array of the **RecordSet** (section 2.2.1.9), returned in the **Result** element of the **ServiceResult** element from a previous call to the **GetData** operation (section 3.1.5.1.1) is "true".

3.1.5.1.2.2 Response Body

Returns the values of the successfully inserted records in the **Values** element of the **RecordSet** (section 2.2.1.9) returned in the **Result** element of the **ServiceResult** (section 2.2.1.11) in the Response Body. In the event of an error on the protocol server, a **ServiceError** (section 2.2.1.10) is returned in the **Error** part of **ServiceResult**.

```
InsertRecordsResult = json-null | serviceResult-member
serviceResult-member = json-quotation-mark "d" json-quotation-mark json-name-
separator ServiceResult
```

InsertRecordsResult: A **ServiceResult** (section 2.2.1.11) that specifies the records which are inserted by the protocol server. MUST be present.

- The **Paging** element of the **RecordSet** MUST be the same as the **Paging** element of the **UpdateRecord** (section 2.2.1.13) which is passed in as an input element in the **Request Body** (section 3.1.5.1.2.1) of the protocol operation.

3.1.5.1.2.3 Processing Details

The protocol server processes **SharedDataBaseInfo** (section 2.2.1.12) and **UpdateRecord** (section 2.2.1.13) and inserts a collection of records in the **Updatable Source** (section 3.1.1), which is specified by **SelectCommand** of **SharedDataBaseInfo**.

The protocol server responds as specified in [MS-ADR] section 3.1.4.8, except as follows:

- The values for the fields of the new records are retrieved from the **NewValues** property of **UpdateRecord**.

- The protocol server retrieves the values of all fields of the newly inserted records from the **Updatable Source** and returns them in the **Values** element of **RecordSet** (section [2.2.1.9](#)), which is the **Result** element of **ServiceResult** (section [2.2.1.11](#)) in Response Body.

In the event of an application error on the protocol server during this operation, the **ServiceError**, as specified in section [2.2.1.10](#), MUST be present in **ServiceResult** (section [2.2.1.11](#)).

3.1.5.1.3 UpdateRecords

This operation updates records in an **Updatable Source** (section [3.1.1](#)), specified in the **SelectCommand** element of **SharedDataBaseInfo** (section [2.2.1.12](#)) in the request body of the operation (section [3.1.5.1.3.1](#)).

The protocol server receives a JSON request that contains the **SharedDataBaseInfo** (section [2.2.1.12](#)) and **UpdateRecord** (section [2.2.1.13](#)) objects. The protocol server processes the request and responds with a JSON response that contains **ServiceResult** (section [2.2.1.11](#)). The **Values** element of **RecordSet** (section [2.2.1.9](#)) returned as the **Result** element of the **ServiceResult** contains the records which the protocol server updated.

In the event of an application error on the protocol server during this operation, the **ServiceError** element (section [2.2.1.10](#)) MUST be present in **ServiceResult** (section [2.2.1.11](#)).

This operation is transported by an HTTP **POST**.

The **serviceName** as specified in section [3.1.5.1](#) MUST be "UpdateRecords".

The processing details for this operation are specified by section [3.1.5.1.3.3](#).

3.1.5.1.3.1 Request Body

The **SharedDataBaseInfo** (section [2.2.1.12](#)) and **UpdateRecord** (section [2.2.1.13](#)) objects in Request Body determine the original and new values for the fields of the records to be updated into the **Updatable Source**. The **Updatable Source** is specified by the **FieldSchema** elements in **RecordSet** (section [2.2.1.9](#)) returned as the **Result** element in the Response Body of a prior **GetData** operation (section [3.1.5.1.1.2](#)).

The original values and new values are specified by the **OriginalValues** and **NewValues** elements of the **UpdateRecord**, respectively. These values MUST be formatted according to the **FormatInfo** element of **SharedDataBaseInfo**. The **SessionId** element of the **SharedDataBaseInfo** MUST be same as the **SessionId** element of the **PagingInfo** element of **RecordSet** (section [2.2.1.9](#)) returned in the **Result** element in Response Body of the first **GetData** operation.

```
UpdateRecordsRequest = dataBaseInfo-member updateRecord-member
dataBaseInfo-member = json-quotation-mark "dataBaseInfo" json-quotation-mark json-name-
separator SharedDataBaseInfo
updateRecord-member = json-quotation-mark "updateRecord" json-quotation-mark json-name-
separator UpdateRecord
```

dataBaseInfo: A **SharedDataBaseInfo** (section [2.2.1.12](#)) that specifies information about the **Updatable Source** to be updated.

updateRecord: An **UpdateRecord** (section [2.2.1.13](#)) that specifies the values for the fields of the records to be updated.

- The number of records in **NewValues** element and **OriginalValues** element MUST be the same.
- The number of fields in each record MUST be same for all the records, and MUST match the number and ordering of fields in **FieldNames** element of **dataBaseInfo**.

- The fields value at an ordinal in a record array from the **OriginalValues** element MUST be of the same data type as specified by the **DataType** element of the **FieldSchema** element at the same ordinal in the **Fields** array of **RecordSet** (section 2.2.1.9), returned in the **Result** element of **ServiceResult** element of a prior call to **GetData** operation (section 3.1.5.1.1).
- Each field value at an ordinal in a record array from the **NewValues** element MUST be formatted according to the **FormatInfo** (section 2.2.1.6) at the same ordinal in the **FormatInfos** array of the **dataBaseInfo**.
- Each primary key fields value at an ordinal in a record array from the **NewValues** element, when converted to a number MUST be equal to the value of field at the same ordinal in a record array from the **OriginalValues** element.
- If value of any field in **OriginalValues** element is "null", that field is ignored by the protocol server and is not updated in the **Updatable Source**.
- The field value at an ordinal in a record array from the **NewValues** and **OriginalValues** elements MUST be "null" if the **ReadOnly** element of the **FieldSchema** element at the same ordinal in the **Fields** array of the **RecordSet** (section 2.2.1.9), returned in the **Result** element of the **ServiceResult** element from a prior call to the **GetData** operation (section 3.1.5.1.1) is "true".

3.1.5.1.3.2 Response Body

Returns the values of the successfully updated records in the **Values** element of the **RecordSet** (section 2.2.1.9) returned in the **Result** element of the **ServiceResult** (section 2.2.1.11) in the Response Body. In the event of an error on the protocol server, a **ServiceError** (section 2.2.1.10) is returned in the **Error** part of **ServiceResult**.

```
UpdateRecordsResult = json-null | serviceResult-member
serviceResult-member = json-quotation-mark "d" json-quotation-mark json-name-
separator ServiceResult
```

UpdateRecordsResult: A **ServiceResult** (section 2.2.1.11) that specifies the records which are updated by the protocol server. MUST be present.

- The **Paging** element of the **RecordSet** MUST be same as the **Paging** element of the **UpdateRecord** (section 2.2.1.13) which is passed in as an input element in the **Request Body** (section 3.1.5.1.3.1) of the protocol operation.

3.1.5.1.3.3 Processing Details

The protocol server processes the **SharedDataBaseInfo** (section 2.2.1.12) and **UpdateRecord** (section 2.2.1.13) and updates a record in the **Updatable Source** (section 3.1.1), specified by **SelectCommand** of **SharedDataBaseInfo**.

The protocol server responds as specified in [MS-ADR] section 3.1.4.12, except as follows:

- The new values for the fields of the records to be updated are retrieved from the **NewValues** property of **UpdateRecord**.
- The original values for the fields of the records to be updated are retrieved from the **OriginalValues** property of **UpdateRecord**.
- The protocol server retrieves the newly updated records from the **Updatable Source** and returns them in the **Values** element of **RecordSet** (section 2.2.1.9) which is returned as the **Result** element of **ServiceResult** (section 2.2.1.11) in Response Body.

3.1.5.1.4 DeleteRecords

This operation deletes records from an **Updatable Source** (section [3.1.1](#)), specified in the **SelectCommand** element of **SharedDataBaseInfo** (section [2.2.1.12](#)) in the request body of the operation (section [3.1.5.1.4.1](#)).

The protocol server receives a JSON request that contains the **SharedDataBaseInfo** (section [2.2.1.12](#)) and **UpdateRecord** (section [2.2.1.13](#)) objects. The protocol server processes the request and responds with a JSON response that contains **ServiceResult** (section [2.2.1.11](#)). The **Values** element of **RecordSet** (section [2.2.1.9](#)) returned as the **Result** element of the **ServiceResult** contains a subset of the remaining records from the **Updatable Source**, after the requested records have been deleted.

In the event of an application error on the protocol server during this operation, the **ServiceError** element (section [2.2.1.10](#)) MUST be present in **ServiceResult** (section [2.2.1.11](#))

This operation is transported by an HTTP **POST**.

The **serviceName** as specified in section [3.1.5.1](#) MUST be "DeleteRecords".

The processing details for this operation are specified by section [3.1.5.1.4.3](#).

3.1.5.1.4.1 Request Body

The **SharedDataBaseInfo** (section [2.2.1.12](#)) and **UpdateRecord** (section [2.2.1.13](#)) objects in Request Body determine the records to be deleted from **Updatable Source** (section [3.1.1](#)). The **Updatable Source** is specified by the **FieldSchema** elements in **RecordSet** (section [2.2.1.9](#)) returned as the **Result** element in the Response Body of a prior **GetData** operation (section [3.1.5.1.1.2](#)).

The **SessionId** element of the **SharedDataBaseInfo** MUST be same as the **SessionId** element of the **PagingInfo** element of **RecordSet**, which is returned in the **Result** element in Response Body of the first **GetData** operation.

```
DeleteRecordsRequest = dataBaseInfo-member updateRecord-member
dataBaseInfo-member = json-quotation-mark "dataBaseInfo" json-quotation-mark json-name-
separator SharedDataBaseInfo
updateRecord-member = json-quotation-mark "updateRecord" json-quotation-mark json-name-
separator UpdateRecord
```

dataBaseInfo: A **SharedDataBaseInfo** (section [2.2.1.12](#)) that specifies information about the **Updatable Source** from which records are to be deleted.

updateRecord: An **UpdateRecord** (section [2.2.1.13](#)) that specifies the original values for the fields of the records to be deleted.

- The number of fields in each record from **OriginalValues** element of **UpdateRecord** MUST be the same for all the records, and MUST match the number and ordering of fields in the **FieldNames** element of **dataBaseInfo**.
- The value of primary key fields in **OriginalValues** element MUST NOT be "null".
- The values of fields which are not primary key, are ignored.
- The **NewValues** element of the **UpdateRecord** MUST be ignored.

3.1.5.1.4.2 Response Body

Returns the values of a subset of remaining records from the **Updatable Source** (section [3.1.1](#)) in the **Values** element of the **RecordSet** (section [2.2.1.9](#)) returned by the **Result** element of the **ServiceResult** (section [2.2.1.11](#)) in the Response Body. In the event of an error on the protocol server, a **ServiceError** (section [2.2.1.10](#)) is returned in the **Error** part of **ServiceResult**.

```
DeleteRecordsResult = json-null | serviceResult-member
serviceResult-member = json-quotation-mark "d" json-quotation-mark json-name-
separator ServiceResult
```

DeleteRecordsResult: A **ServiceResult** (section [2.2.1.11](#)) that specifies a subset of the remaining records from **Updatable Source** after the requested records have been deleted by protocol server. **MUST** be present.

- The **Paging** element of the **RecordSet** **MUST** be same as the **Paging** element of the **UpdateRecord** (section [2.2.1.13](#)) which is passed in as an input element in the **Request Body** (section [3.1.5.1.2.1](#)) of the protocol operation.

3.1.5.1.4.3 Processing Details

The protocol server processes the **SharedDataBaseInfo** (section [2.2.1.12](#)) and **UpdateRecord** (section [2.2.1.13](#)) and then deletes a collection of records from the **Updatable Source** (section [3.1.1](#)), which is specified by **SelectCommand** of **SharedDataBaseInfo**.

The protocol server responds as specified in [\[MS-ADR\]](#) section 3.1.4.2, except as follows:

- The values of the primary key fields of each records are retrieved from the **OriginalValues** element of **UpdateRecord**.
- The protocol server returns a subset of the remaining records retrieved according to the **FirstRow** and the **PageSize** elements of the **PagingInfo** element of the **UpdateRecord**.

3.1.5.1.5 GetDistinctValues

This operation retrieves records with distinct values for a field from **Source** (section [3.1.1](#)), specified in the **SelectCommand** element of **SharedDataBaseInfo** (section [2.2.1.12](#)) in the request body of the operation (section [3.1.5.1.5.1](#)).

The protocol server receives a JSON request that contains the **SharedDataBaseInfo** and **PagingInfo** (section [2.2.1.7](#)) objects. The request also contains a **columnName** for which distinct values are requested. The protocol server processes the request and responds with a JSON response that contains **ServiceResult** (section [2.2.1.11](#)). The **Values** element of **RecordSet** (section [2.2.1.9](#)) returned as the **Result** element of the **ServiceResult** contains the records from the **Source**, which contain distinct values for the requested **columnName**.

In the event of an application error on the protocol server during this operation, the **ServiceError** element (section [2.2.1.10](#)) **MUST** be present in **ServiceResult** (section [2.2.1.11](#)).

This operation is transported by an HTTP **POST**.

The **serviceName** as specified in section [3.1.5.1](#) **MUST** be "GetDistinctValues".

The processing details for this operation are specified by section [3.1.5.1.5.3](#).

3.1.5.1.5.1 Request Body

The **SharedDataBaseInfo** (section [2.2.1.12](#)) and **PagingInfo** (section [2.2.1.7](#)) objects and the **columnName** element in Request Body determine which records are to be fetched from a **Source** (section [3.1.1](#)), which is specified by the **SelectCommand** element of the **SharedDataBaseInfo**.

```
GetDistinctValuesRequest = dataBaseInfo-member columnName-member pagingInfo-member
dataBaseInfo-member = json-quotation-mark "dataBaseInfo" json-quotation-mark json-name-
separator SharedDataBaseInfo
columnName-member = json-quotation-mark "columnName" json-quotation-mark json-name-
separator json-string
pagingInfo-member = json-quotation-mark "pagingInfo" json-quotation-mark json-name-
separator PagingInfo
```

dataBaseInfo: A **SharedDataBaseInfo** (section [2.2.1.12](#)) that specifies the **Source**, the fields from this **Source**, the sort order and restriction criteria of the records that are retrieved from the database application.

columnName: A **json-string** (section [2.2](#)) that specifies the field in the **Source** for which distinct values are retrieved.

pagingInfo: A **PagingInfo** (section [2.2.1.7](#)) that specifies which records are retrieved from the database application.

3.1.5.1.5.2 Response Body

Returns the values of field specified by **columnName** from the **Source** (section [3.1.1](#)) in the **Values** element of the **RecordSet** (section [2.2.1.9](#)) returned by the **Result** element of the **ServiceResult** (section [2.2.1.11](#)) in the Response Body. In the event of an error on the protocol server, a **ServiceError** (section [2.2.1.10](#)) is returned in the **Error** part of **ServiceResult**

```
GetDistinctValuesResult = json-null | serviceResult-member
serviceResult-member = json-quotation-mark "d" json-quotation-mark json-name-
separator ServiceResult
```

GetDistinctValuesResult: A **ServiceResult** (section [2.2.1.11](#)) that specifies the rows containing distinct values for the field specified by **columnName** element. MUST be present.

3.1.5.1.5.3 Processing Details

The protocol server processes the **SharedDataBaseInfo** (section [2.2.1.12](#)) and **PagingInfo** (section [2.2.1.7](#)) objects and returns a subset of records from the **Source** (section [3.1.1](#)), which contains distinct values for the field, specified by **columnName** in the Request Body.

The protocol server responds as specified in [\[MS-ADR\]](#) section 3.1.4.5, except as follows:

- The protocol server retrieves data from the given **Source** passed in as the **SelectCommand** element of the **SharedDataBaseInfo** (section [2.2.1.12](#)), an input element of the Request Body (section [3.1.5.1.5.1](#)).
- Values are retrieved only for the field that is specified by **columnName** input element of the Request Body.

3.1.5.1.6 FixupRow

This operation returns the resulting values of modeling a hypothetical update on one of the records in an **Updatable Source** (section [3.1.1](#)), specified in the **SelectCommand** element of **SharedDataBaseInfo** (section [2.2.1.12](#)) in the request body of the operation (section [3.1.5.1.6.1](#)).

The protocol server receives a JSON request that contains the **SharedDataBaseInfo** (section 2.2.1.12) and **FixupRecord** (section 2.2.1.5) objects. The protocol server processes the request and responds with a JSON response that contains **ServiceResult** (section 2.2.1.11). The **Values** element of **RecordSet** (section 2.2.1.9) returned as **Result** element of the **ServiceResult** contains the record from the **Updatable Source**, which contains resulting values of hypothetical update on that record.

In the event of an application error on the protocol server during this operation, the **ServiceError** element (section 2.2.1.10) MUST be present in **ServiceResult** (section 2.2.1.11)

This operation is transported by an HTTP **POST**.

The **serviceName** as specified in section 3.1.5.1 MUST be "FixupRow".

The processing details for this operation are specified by section 3.1.5.1.6.3.

3.1.5.1.6.1 Request Body

The **SharedDataBaseInfo** (section 2.2.1.12) and **FixupRecord** (section 2.2.1.13) objects in Request Body determine the values for the fields of the record for which an update is to be modeled in the **Updatable Source** (section 3.1.1). The **Updatable Source** is specified by the **FieldSchema** elements in **RecordSet** (section 2.2.1.9) returned as the **Result** element in the Response Body of a prior **GetData** operation (section 3.1.5.1.1.2). This operation just returns the result of the hypothetical update. The fields in **Updatable Source** are not updated.

```
FixupRowRequest = dataBaseInfo-member fixupRecord-member
dataBaseInfo-member = json-quotation-mark "dataBaseInfo" json-quotation-mark json-name-
separator SharedDataBaseInfo
fixupRecord-member = json-quotation-mark "fixupRecord" json-quotation-mark json-name-
separator FixupRecord
```

dataBaseInfo: A **SharedDataBaseInfo** (section 2.2.1.12) that specifies information about **Updatable Source** on which hypothetical update is to be performed.

- The **SessionId** element of **SharedDataBaseInfo** MUST be same as the **SessionId** element of the **PagingInfo** element of **RecordSet** (section 2.2.1.9), which is returned in the **Result** element in Response Body of the first **GetData** operation (section 3.1.5.1.1.2).

fixupRecord: A **FixupRecord** (section 2.2.1.5) that specifies the values for the fields of the records which are to be used for the hypothetical update.

- The values to be updated are specified by the **SupportingFieldValues** element of **FixupRecord**.
- These values are for the fields whose indexes in **Fields** array of **RecordSet** (section 2.2.1.9), returned in the **Result** element of **ServiceResult** element of a previous call to **GetData** operation (section 3.1.5.1.1) are specified by **SupportingFieldIndexes** element.
- These values MUST be formatted according to the **FormatInfo** element of **SharedDataBaseInfo**.
- The number of values in **SupportingFieldValues** element and **SupportingFieldIndexes** element MUST be the same.
- The value of the primary key field of the **Updatable Source** is specified by the **Key** element of the **FixupRecord**.

3.1.5.1.6.2 Response Body

Returns the resulting values of the successful hypothetical update in the **Values** element of the **RecordSet** (section 2.2.1.9) returned in the **Result** element of the **ServiceResult** (section 2.2.1.11)

in the Response Body. In the event of an error on the protocol server, a **ServiceError** (section [2.2.1.10](#)) is returned in the **Error** part of **ServiceResult**.

```
FixupRowResult = json-null | serviceResult-member
serviceResult-member = json-quotation-mark "d" json-quotation-mark json-name-
separator ServiceResult
```

FixupRowResult: A **ServiceResult** (section 2.2.1.11) that specifies the row that is the result of hypothetical update. MUST be present.

3.1.5.1.6.3 Processing Details

The protocol server processes the **SharedDataBaseInfo** (section [2.2.1.12](#)) and **FixupRecord** (section [2.2.1.5](#)) objects and returns a record from the **Updatable Source** (section [3.1.1](#)), which is specified by the **SelectCommand** element of **SharedDataBaseInfo** (section 2.2.1.12). The values contain the resulting values of a hypothetical update performed on that record.

The protocol server responds as specified in [\[MS-ADR\]](#) section 3.1.4.3, except as follows:

- The original record is retrieved from **Updatable Source** (section 3.1.1). The primary key of the record MUST match the **Key** element from **FixupRecord** (section 2.2.1.5).
- In the event of an application error on the protocol server during this operation, the **ServiceError**, as specified in section [2.2.1.10](#), MUST be present in **ServiceResult** (section [2.2.1.11](#)).

3.1.5.1.7 GetSearchData

This operation retrieves records from **Source** (section [3.1.1](#)), specified in the **SelectCommand** element of **SharedDataBaseInfo** (section [2.2.1.12](#)) in the request body of the operation (section [3.1.5.1.7.1](#)).

The protocol server receives a JSON request that contains the **SharedDataBaseInfo** object. The protocol server processes the request and responds with a JSON response that contains **ServiceResult** (section [2.2.1.11](#)). The **Values** element of **RecordSet** (section [2.2.1.9](#)) returned as the **Result** element of the **ServiceResult** contains the records from the **Source**, which matches the following conditions.

- A maximum of 9 records are retrieved.
- In each record, the combination of values of fields whose indexes in **FieldNames** are 0 and 1 MUST be distinct.
- Only records where value of the field whose index in **FieldNames** is 1, contains the **Restriction** element of **SharedDataBaseInfo** are retrieved.

In the event of an application error on the protocol server during this operation, the **ServiceError** element (section [2.2.1.10](#)) MUST be present in **ServiceResult** (section 2.2.1.11).

This operation is transported by an HTTP **POST**.

The **serviceName** as specified in section [3.1.5.1](#) MUST be "GetSearchData".

The processing details for this operation are specified by section [3.1.5.1.7.3](#).

3.1.5.1.7.1 Request Body

The **SharedDataBaseInfo** (section [2.2.1.12](#)) object in Request Body determines fields to be retrieved from a **Source** (section [3.1.1](#)), which is specified by the **SelectCommand** element of the **SharedDataBaseInfo**.

```
GetSearchDataRequest = dataBaseInfo-member  
dataBaseInfo-member = json-quotation-mark "dataBaseInfo" json-quotation-mark json-name-  
separator SharedDataBaseInfo
```

dataBaseInfo: A **SharedDataBaseInfo** (section [2.2.1.12](#)) that specifies information about **Source** from which records are to be retrieved from database application.

- The fields for which data is to be retrieved are specified by the **FieldNames** element of the **SharedDataBaseInfo**.
- At least two fields MUST be present in the **FieldNames** element.
- At most three fields MUST be present in the **FieldNames** element.
- **Restriction** element of **SharedDataBaseInfo** MUST NOT be empty.

3.1.5.1.7.2 Response Body

Returns the values of field specified by the **FieldNames** element of **SharedDataBaseInfo** (section [2.2.1.12](#)) from the **Source** (section [3.1.1](#)) in the **Values** element of the **RecordSet** (section [2.2.1.9](#)) returned by the **Result** element of the **ServiceResult** (section [2.2.1.11](#)) in the Response Body. In the event of an error on the protocol server, a **ServiceError** (section [2.2.1.10](#)) is returned in the **Error** part of **ServiceResult**.

```
GetSearchDataResult = json-null | serviceResult-member  
serviceResult-member = json-quotation-mark "d" json-quotation-mark json-name-  
separator ServiceResult
```

GetSearchDataResult: A **ServiceResult** (section [2.2.1.11](#)) that specifies the rows containing values for field specified by **FieldNames** element of **SharedDataBaseInfo**. MUST be present.

3.1.5.1.7.3 Processing Details

The protocol server processes the **SharedDataBaseInfo** (section [2.2.1.12](#)) object and returns at most 9 records from the **Source** (section [3.1.1](#)), which contains distinct values for the combination of fields, with indexes 0 and 1 in the **FieldNames** element of **SharedDataBaseInfo** element in the Request Body.

The processing is done as follows.

- The protocol server retrieves data from the given **Source** passed in as the **SelectCommand** element of the **SharedDataBaseInfo** (section [2.2.1.12](#)), an input element of the Request Body (section [3.1.5.1.5.1](#)).
- Values are retrieved only for the first three fields specified by the **FieldNames** element of the **SharedDataBaseInfo** input element of the Request Body.
- In each record, the combination of values of fields whose indexes in **FieldNames** are 0 and 1 MUST be distinct.
- Only records where value of the field whose index in **FieldNames** is 1, contains the **Restriction** element of **SharedDataBaseInfo** are retrieved.

3.1.6 Timer Events

3.1.6.1 Session Timeout

Session Timeout, as specified by [\[MS-ADR\]](#) section 3.1.5.1.

3.1.7 Other Local Events

None.

4 Protocol Examples

4.1 GetData Service Operation

This example shows fetching a subset of rows from a **Source** (section [3.1.1](#)).

The protocol client sends the following message to the protocol server to perform a **GetData** operation.

```
{ "dataBaseInfo" : { "AllowAdditions" : true,
  "AllowDeletions" : true,
  "AllowEdits" : true,
  "FetchSchema" : true,
  "SelectCommand" : "Employees"
},
  "pagingInfo" : { "CacheCommands" : 8,
  "FirstRow" : 0,
  "PageSize" : 50,
  "SortExpression" : "<Ordering
xmlns='http://schemas.microsoft.com/office/accessservices/2010/12/application\'
/><Order Name='FirstName' Direction='Ascending' /></Ordering>",
  "UseCache" : false
}
}
```

The protocol server sends back the following response.

```
{ "d" : { "Error" : null,
  "Result" : { "AggregateLocalized" : null,
  "AggregateValues" : null,
  "AnonymousCanInsert" : false,
  "AnonymousCanRead" : true,
  "DataMacroInstanceIds" : null,
  "Fields" : [ { "AllowMultipleValues" : false,
    "ColumnName" : "ID",
    "CurrencySymbol" : null,
    "DataType" : "Int",
    "DecimalPlaces" : -1,
    "DefaultExpression" : null,
    "DefaultValue" : "",
    "DependentFields" : null,
    "FormatString" : null,
    "IsKey" : true,
    "IsTableQueryLookup" : false,
    "KeyIndex" : -1,
    "LookupBoundField" : null,
    "LookupDisplayField" : null,
    "LookupSortType" : null,
    "LookupSource" : null,
    "MaxLength" : 4,
    "ReadOnly" : true,
    "Required" : true,
    "SourceObject" : null,
    "TextType" : null,
    "ValidationMessage" : null,
    "ValidationScript" : null
  },
  { "AllowMultipleValues" : false,
```

```

        "ColumnName" : "FirstName",
        "CurrencySymbol" : null,
        "DataType" : "NVarChar",
        "DecimalPlaces" : -1,
        "DefaultExpression" : null,
        "DefaultValue" : "",
        "DependentFields" : null,
        "FormatString" : null,
        "IsKey" : false,
        "IsTableQueryLookup" : false,
        "KeyIndex" : -1,
        "LookupBoundField" : null,
        "LookupDisplayField" : null,
        "LookupSortType" : null,
        "LookupSource" : null,
        "MaxLength" : 220,
        "ReadOnly" : false,
        "Required" : false,
        "SourceObject" : null,
        "TextType" : "SingleLine",
        "ValidationMessage" : null,
        "ValidationScript" : null
    },
    { "AllowMultipleValues" : false,
      "ColumnName" : "LastName",
      "CurrencySymbol" : null,
      "DataType" : "NVarChar",
      "DecimalPlaces" : -1,
      "DefaultExpression" : null,
      "DefaultValue" : "",
      "DependentFields" : null,
      "FormatString" : null,
      "IsKey" : false,
      "IsTableQueryLookup" : false,
      "KeyIndex" : -1,
      "LookupBoundField" : null,
      "LookupDisplayField" : null,
      "LookupSortType" : null,
      "LookupSource" : null,
      "MaxLength" : 220,
      "ReadOnly" : false,
      "Required" : false,
      "SourceObject" : null,
      "TextType" : "SingleLine",
      "ValidationMessage" : null,
      "ValidationScript" : null
    }
  ],
  "FormatInfos" : [ [ { "Currency" : null,
    "Format" : null,
    "Precision" : -1,
    "__type" : "Microsoft.Office.Access.Services.UI.FormatInfo"
  } ],
  [ { "Currency" : null,
    "Format" : null,
    "Precision" : -1,
    "__type" : "Microsoft.Office.Access.Services.UI.FormatInfo"
  } ],
  [ { "Currency" : null,
    "Format" : null,
    "Precision" : -1,

```

```

        "__type" : "Microsoft.Office.Access.Services.UI.FormatInfo"
    } ]
  ],
  "IsAnonymous" : false,
  "Localized" : [ [ null,
    null,
    null
  ] ],
  "Paging" : { "CacheCommands" : 8,
    "Filter" : null,
    "FirstRow" : 0,
    "Moniker" : null,
    "PageSize" : 50,
    "SessionId" : "36.31b40d30-46c9-4850-9546-
3ac936024abd163.1.V22.50DEXMZ5ML2xUiPMVnvjBS90.5.en-US5.en-US73.+0480#0000-11-00-
01T02:00:00:0000#+0000#0000-03-00-02T02:00:00:0000#+-006036.00000000-0000-0000-0000-
0000000000001.U",
    "SortExpression" : "<Ordering
xmlns=\\"http://schemas.microsoft.com/office/accessservices/2010/12/application\\"
><Order Name=\\"FirstName\\" Direction=\\"Ascending\\" /></Ordering>",
    "TotalRows" : 1,
    "UseCache" : false,
    "__type" : "Microsoft.Office.Access.Services.UI.PagingInfo"
  },
  "RecordStatuses" : null,
  "Values" : [ [ 1,
    "Updated First Name",
    "Last Name"
  ] ]
},
  "__type" : "Microsoft.Office.Access.Services.UI.ServiceResult"
} }

```

4.2 InsertRecords Service Operation

This example describes how to use the **InsertRecords** Service Operation method (section [3.1.5.1.2](#)) to insert an item into a table.

The protocol client sends the following message to the protocol server to insert an item into a table.

```

{ "dataBaseInfo" : { "AggregateExpressions" : null,
  "AggregateFormatInfos" : [ ],
  "AllowAdditions" : true,
  "AllowDeletions" : true,
  "AllowEdits" : true,
  "FetchSchema" : false,
  "FieldNames" : [ "ID",
    "FirstName",
    "LastName"
  ],
  "FormatInfos" : [ [ { "Currency" : null,
    "Format" : null,
    "Precision" : -1
  } ] ],
  "ParameterValues" : null,
  "SelectCommand" : "Employees",
  "SessionId" : "36.31b40d30-46c9-4850-9546-
3ac936024abd163.1.V22.47oNwJCV0bA0inRQRXct3u90.5.en-US5.en-US73.+0480#0000-11-00-

```

```

01T02:00:00:0000#+0000#0000-03-00-02T02:00:00:0000#-006036.00000000-0000-0000-0000-
0000000000001.U"
  },
  "updateRecord" : { "NewValues" : [ [ null,
    "First Name",
    "Last Name"
  ] ],
  "Paging" : { "CacheCommands" : 0,
    "Filter" : null,
    "FirstRow" : 0,
    "Moniker" : null,
    "PageSize" : 1,
    "SessionId" : null,
    "SortExpression" : "<Ordering
xmlns=\\"http://schemas.microsoft.com/office/accessservices/2010/12/application\\"
><Order Name=\\"FirstName\\" Direction=\\"Ascending\\" /></Ordering>",
    "TotalRows" : 1,
    "UseCache" : true
  },
  "ReturnDataMacroIds" : false
}
}

```

The protocol server sends back the following response.

```

{ "d" : { "Error" : null,
  "Result" : { "AggregateLocalized" : null,
    "AggregateValues" : null,
    "AnonymousCanInsert" : false,
    "AnonymousCanRead" : true,
    "DataMacroInstanceIds" : null,
    "Fields" : null,
    "FormatInfos" : [ [ { "Currency" : null,
      "Format" : null,
      "Precision" : -1,
      "__type" : "Microsoft.Office.Access.Services.UI.FormatInfo"
    } ],
    [ { "Currency" : null,
      "Format" : null,
      "Precision" : -1,
      "__type" : "Microsoft.Office.Access.Services.UI.FormatInfo"
    } ],
    [ { "Currency" : null,
      "Format" : null,
      "Precision" : -1,
      "__type" : "Microsoft.Office.Access.Services.UI.FormatInfo"
    } ]
  ],
  "IsAnonymous" : false,
  "Localized" : [ [ null,
    null,
    null
  ] ],
  "Paging" : { "CacheCommands" : 0,
    "Filter" : null,
    "FirstRow" : 0,
    "Moniker" : null,
    "PageSize" : 1,
    "SessionId" : null,

```

```

        "SortExpression" : "<Ordering
xmlns=\"http://schemas.microsoft.com/office/accessservices/2010/12/application\"
><Order Name=\"FirstName\" Direction=\"Ascending\" /></Ordering>",
        "TotalRows" : 1,
        "UseCache" : true,
        "__type" : "Microsoft.Office.Access.Services.UI.PagingInfo"
    },
    "RecordStatuses" : null,
    "Values" : [ [ 1,
        "First Name",
        "Last Name"
    ] ]
},
    "__type" : "Microsoft.Office.Access.Services.UI.ServiceResult"
} }

```

4.3 UpdateRecords Service Operation

This example describes how to use the **UpdateRecords** Service Operation method (section [3.1.5.1.3](#)) to update an item into a table.

The protocol client sends the following message to the protocol server to update an item into a table:

```

{ "dataBaseInfo" : { "AggregateExpressions" : null,
    "AggregateFormatInfos" : [ ],
    "AllowAdditions" : true,
    "AllowDeletions" : true,
    "AllowEdits" : true,
    "FetchSchema" : false,
    "FieldNames" : [ "ID",
        "FirstName",
        "LastName"
    ],
    "FormatInfos" : [ [ { "Currency" : null,
        "Format" : null,
        "Precision" : -1
    } ] ],
    "ParameterValues" : null,
    "SelectCommand" : "Employees",
    "SessionId" : "36.31b40d30-46c9-4850-9546-
3ac936024abd163.1.V22.47oNwJCV0bA0inRQRXct3u90.5.en-US5.en-US73.+0480#0000-11-00-
01T02:00:00:0000#+0000#0000-03-00-02T02:00:00:0000#+006036.00000000-0000-0000-0000-
0000000000001.U"
},
    "updateRecord" : { "NewValues" : [ [ "1",
        "Updated First Name",
        null
    ] ],
    "OriginalValues" : [ [ 1,
        "First Name",
        null
    ] ],
    "Paging" : { "CacheCommands" : 0,
        "Filter" : null,
        "FirstRow" : 0,
        "Moniker" : null,
        "PageSize" : 1,
        "SessionId" : null,

```

```

        "SortExpression" : "<Ordering
xmlns=\\"http://schemas.microsoft.com/office/accessservices/2010/12/application\\"
><Order Name=\\"FirstName\\" Direction=\\"Ascending\\" /></Ordering>",
        "TotalRows" : 1,
        "UseCache" : true,
        "__type" : "Microsoft.Office.Access.Services.UI.PagingInfo"
    },
    "ReturnDataMacroIds" : false
}
}
}

```

The protocol server sends back the following response.

```

{ "d" : { "Error" : null,
  "Result" : { "AggregateLocalized" : null,
    "AggregateValues" : null,
    "AnonymousCanInsert" : false,
    "AnonymousCanRead" : true,
    "DataMacroInstanceIds" : null,
    "Fields" : null,
    "FormatInfos" : [ [ { "Currency" : null,
      "Format" : null,
      "Precision" : -1,
      "__type" : "Microsoft.Office.Access.Services.UI.FormatInfo"
    } ],
    [ { "Currency" : null,
      "Format" : null,
      "Precision" : -1,
      "__type" : "Microsoft.Office.Access.Services.UI.FormatInfo"
    } ],
    [ { "Currency" : null,
      "Format" : null,
      "Precision" : -1,
      "__type" : "Microsoft.Office.Access.Services.UI.FormatInfo"
    } ]
  ],
  "IsAnonymous" : false,
  "Localized" : [ [ null,
    null,
    null
  ] ],
  "Paging" : { "CacheCommands" : 0,
    "Filter" : null,
    "FirstRow" : 0,
    "Moniker" : null,
    "PageSize" : 1,
    "SessionId" : null,
    "SortExpression" : "<Ordering
xmlns=\\"http://schemas.microsoft.com/office/accessservices/2010/12/application\\"
><Order Name=\\"FirstName\\" Direction=\\"Ascending\\" /></Ordering>",
    "TotalRows" : 1,
    "UseCache" : true,
    "__type" : "Microsoft.Office.Access.Services.UI.PagingInfo"
  },
  "RecordStatuses" : null,
  "Values" : [ [ 1,
    "Updated First Name",
    "Last Name"
  ] ]
}

```

```

    ] ]
  },
  "__type" : "Microsoft.Office.Access.Services.UI.ServiceResult"
} }

```

4.4 DeleteRecords Service Operation

This example describes how to use the **DeleteRecords** Service Operation method (section [3.1.5.1.4](#)) to delete an item from a table.

The protocol client sends the following message to the protocol server to delete an item from a table.

```

{ "dataBaseInfo" : { "AggregateExpressions" : null,
  "AggregateFormatInfos" : [ ],
  "AllowAdditions" : true,
  "AllowDeletions" : true,
  "AllowEdits" : true,
  "FetchSchema" : false,
  "FieldNames" : [ "ID",
    "FirstName",
    "LastName"
  ],
  "FormatInfos" : [ [ { "Currency" : null,
    "Format" : null,
    "Precision" : -1
  } ] ],
  "ParameterValues" : null,
  "SelectCommand" : "Employees",
  "SessionId" : "36.31b40d30-46c9-4850-9546-3ac936024abd163.1.V22.51RMVPQFdeHSB2Ag5Mm53S90.5.en-US5.en-US73.+0480#0000-11-00-01T02:00:00:0000#+0000#0000-03-00-02T02:00:00:0000#-006036.00000000-0000-0000-0000-00000000000001.U"
},
  "updateRecord" : { "OriginalValues" : [ [ 2,
    "First Name 1",
    "Last Name 1"
  ] ],
  "Paging" : { "CacheCommands" : 0,
    "Filter" : null,
    "FirstRow" : 1,
    "Moniker" : null,
    "PageSize" : 1,
    "SessionId" : null,
    "SortExpression" : "<Ordering xmlns='http://schemas.microsoft.com/office/accessservices/2010/12/application'><Order Name='FirstName' Direction='Ascending' /></Ordering>",
    "TotalRows" : 2,
    "UseCache" : true,
    "__type" : "Microsoft.Office.Access.Services.UI.PagingInfo"
  },
  "ReturnDataMacroIds" : false
}
}

```

The protocol server sends back the following response.

```

{ "d" : { "Error" : null,

        "Result" : { "AggregateLocalized" : null,
                    "AggregateValues" : null,
                    "AnonymousCanInsert" : false,
                    "AnonymousCanRead" : true,
                    "DataMacroInstanceIds" : null,
                    "Fields" : null,
                    "FormatInfos" : [ [ { "Currency" : null,
                                         "Format" : null,
                                         "Precision" : -1,
                                         "__type" : "Microsoft.Office.Access.Services.UI.FormatInfo"
                                       } ],
                                     [ { "Currency" : null,
                                         "Format" : null,
                                         "Precision" : -1,
                                         "__type" : "Microsoft.Office.Access.Services.UI.FormatInfo"
                                       } ],
                                     [ { "Currency" : null,
                                         "Format" : null,
                                         "Precision" : -1,
                                         "__type" : "Microsoft.Office.Access.Services.UI.FormatInfo"
                                       } ]
                                      ],
                    "IsAnonymous" : false,
                    "Localized" : [ [ null,
                                     null,
                                     null
                                   ] ],
                    "Paging" : { "CacheCommands" : 0,
                                 "Filter" : null,
                                 "FirstRow" : 0,
                                 "Moniker" : null,

```



```
    "PageSize" : 1,
    "SessionId" : "36.31b40d30-46c9-4850-9546-3ac936024abd163.1.V22.51RMVPQFdeHSB2Ag5Mm53S90.5.en-US5.en-US73.+0480#0000-11-00-01T02:00:00:0000#+0000#0000-03-00-02T02:00:00:0000#+006036.00000000-0000-0000-0000-0000000000001.U",
    "SortExpression" : "<Ordering xmlns=\"http://schemas.microsoft.com/office/accessservices/2010/12/application\"><Order Name=\"FirstName\" Direction=\"Ascending\" /></Ordering>",
    "TotalRows" : 1,
    "UseCache" : true,
    "__type" : "Microsoft.Office.Access.Services.UI.PagingInfo"
  },
  "RecordStatuses" : null,
  "Values" : [ [ 1,
    "First Name",
    "Last Name"
  ] ]
},
 "__type" : "Microsoft.Office.Access.Services.UI.ServiceResult"
} }
```

5 Security

5.1 Security Considerations for Implementers

In addition to the security considerations applicable to the underlying protocols, there are security risks associated with exposing **session identifiers**. If a session identifier is exposed, it is possible for an attacker to read information from, or modify data in, a session on the protocol server. An implementer of this protocol needs to consider keeping session identifiers protected. There could be some cases where it is desirable to expose a session identifier, however, an implementer is to use caution in how they expose session identifiers and consider the security risks.

5.2 Index of Security Parameters

None.

6 Appendix A: Full JSON ABNF

```
ClientMessage = json-object
MessageID = json-quotation-mark "MessageID" json-quotation-mark json-name-
separator ClientMessageID
Context = json-quotation-mark "Context" json-quotation-mark json-name-separator
JsonArrayOfAnyType

CurrentUserPermissions = json-object
Read = json-quotation-mark "Read" json-quotation-mark json-name-separator json-
bool
Write = json-quotation-mark "Write" json-quotation-mark json-name-separator json-
bool
Author = json-quotation-mark "Author" json-quotation-mark json-name-separator
json-bool
IsAuthenticated = json-quotation-mark "IsAuthenticated" json-quotation-mark json-
name-separator json-bool
FieldSchema = json-object
ColumnName = json-quotation-mark "ColumnName" json-quotation-mark json-name-
separator json-string
DataType = json-quotation-mark "DataType" json-quotation-mark json-name-
separator json-string
DefaultValue = json-quotation-mark "DefaultValue" json-quotation-mark json-name-
separator json-string
IsKey = json-quotation-mark "IsKey" json-quotation-mark json-name-separator json-
bool
Required = json-quotation-mark "Required" json-quotation-mark json-name-
separator json-bool
ReadOnly = json-quotation-mark "ReadOnly" json-quotation-mark json-name-
separator json-bool
MaxLength = json-quotation-mark "MaxLength" json-quotation-mark json-name-
separator json-int
DefaultExpression = json-quotation-mark "DefaultExpression" json-quotation-mark
json-name-separator json-string
ValidationScript = json-quotation-mark "ValidationScript" json-quotation-mark json-
name-separator json-string
ValidationMessage = json-quotation-mark "ValidationMessage" json-quotation-mark
json-name-separator ClientMessage
KeyIndex = json-quotation-mark "KeyIndex" json-quotation-mark json-name-
separator json-int
SourceObject = json-quotation-mark "SourceObject" json-quotation-mark json-name-
separator json-string
DependentFields = json-quotation-mark "DependentFields" json-quotation-mark json-
name-separator JsonArrayOfInt
AllowMultipleValues = json-quotation-mark "AllowMultipleValues" json-quotation-mark
json-name-separator json-bool
FormatString = json-quotation-mark "FormatString" json-quotation-mark json-name-
separator json-string
CurrencySymbol = json-quotation-mark "CurrencySymbol" json-quotation-mark json-
name-separator json-string
DecimalPlaces = json-quotation-mark "DecimalPlaces" json-quotation-mark json-name-
separator json-int
TextType = json-quotation-mark "TextType" json-quotation-mark json-name-
separator json-string
IsTableQueryLookup = json-quotation-mark "IsTableQueryLookup" json-quotation-mark
json-name-separator json-bool
LookupSource = json-quotation-mark "LookupSource" json-quotation-mark json-name-
separator json-string
LookupBoundField = json-quotation-mark "LookupBoundField" json-quotation-mark json-
name-separator json-string
```

```

LookupDisplayField = json-quotation-mark "LookupDisplayField" json-quotation-mark
json-name-separator json-string

FilterInfo = json-object
Culture = json-quotation-mark "Culture" json-quotation-mark json-name-separator
json-string
Expression = json-quotation-mark "Expression" json-quotation-mark json-name-
separator json-string
Fields = json-quotation-mark "Fields" json-quotation-mark json-name-separator
JsonArrayOfStrings
Text = json-quotation-mark "Text" json-quotation-mark json-name-separator json-
string

FixupRecord = json-object
Key = json-quotation-mark "Key" json-quotation-mark json-name-separator json-string
SupportingFieldIndexes = json-quotation-mark "SupportingFieldIndexes" json-
quotation-mark json-name-separator JsonArrayOfInt
SupportingFieldValues = json-quotation-mark "SupportingFieldValues" json-quotation-
mark json-name-separator JsonArrayOfString

FormatInfo = json-object
Currency = json-quotation-mark "Currency" json-quotation-mark json-name-separator
json-string
Format = json-quotation-mark "Format" json-quotation-mark json-name-separator json-
string
Precision = json-quotation-mark "Precision" json-quotation-mark json-name-separator
json-int

PagingInfo = json-object
FirstRow = json-quotation-mark "FirstRow" json-quotation-mark json-name-separator
json-int
PageSize = json-quotation-mark "PageSize" json-quotation-mark json-name-separator
json-int
Moniker = json-quotation-mark "Moniker" json-quotation-mark json-name-separator
(json-string / json-null)
UseCache = json-quotation-mark "UseCache" json-quotation-mark json-name-separator
json-bool
CacheCommands = json-quotation-mark "CacheCommands" json-quotation-mark json-name-
separator CacheCommands
SortExpression = json-quotation-mark "SortExpression" json-quotation-mark json-
name-separator json-string
Filter = json-quotation-mark "Filter" json-quotation-mark json-name-separator
FilterInfo
TotalRows = json-quotation-mark "TotalRows" json-quotation-mark json-name-separator
json-int
SessionId = json-quotation-mark "SessionId" json-quotation-mark json-name-separator
json-string
RetrieveExactRowCount = json-quotation-mark "RetrieveExactRowCount" json-quotation-
mark json-name-separator json-bool
RowKey = json-quotation-mark "RowKey" json-quotation-mark json-name-separator json-
int

ParameterValue = json-object
Name = json-quotation-mark "Name" json-quotation-mark json-name-separator json-
string
Value = json-quotation-mark "Value" json-quotation-mark json-name-separator json-
value

RecordSet = json-object
Values = json-quotation-mark "Values" json-quotation-mark json-name-separator
JsonArrayOfArrayOfAnyType

```

```

Localized = json-quotation-mark "Localized" json-quotation-mark json-name-separator
JsonArrayOfArrayOfArrayOfString
Paging = json-quotation-mark "Paging" json-quotation-mark json-name-separator
PagingInfo
Fields = json-quotation-mark "Fields" json-quotation-mark json-name-separator
json-begin-array [FieldSchema *( json-value-separator FieldSchema) ] json-end-array
FormatInfos = json-quotation-mark "FormatInfos" json-quotation-mark json-name-
separator json-begin-array [FormatInfo *( json-value-separator FormatInfo) ] json-
end-array
AutoSumValues = json-quotation-mark "AutoSumValues" json-quotation-mark json-name-
separator jsonArrayOfAnyType
CurrentUserPermissions = json-quotation-mark "CurrentUserPermissions" json-
quotation-mark json-name-separator CurrentUserPermissions
RelatedFieldInfos = json-quotation-mark "RelatedFieldInfos" json-quotation-mark
json-name-separator json-begin-array [FieldSchema *( json-value-separator
FieldSchema) ] json-end-array

ServiceError = json-object
Message = json-quotation-mark "Message" json-quotation-mark json-name-separator
ClientMessage
Caption = json-quotation-mark "Caption" json-quotation-mark json-name-separator
(json-string / json-null)
HelpText = json-quotation-mark "HelpText" json-quotation-mark json-name-separator
(json-string / json-null)
HelpId = json-quotation-mark "HelpId" json-quotation-mark json-name-separator
(json-string / json-null)
Severity = json-quotation-mark "Severity" json-quotation-mark json-name-separator
MessageSeverity
Number = json-quotation-mark "Number" json-quotation-mark json-name-separator json-
value

ServiceResult = json-object
Result = json-quotation-mark "Result" json-quotation-mark json-name-separator json-
value
Error = json-quotation-mark "Error" json-quotation-mark json-name-separator
ServiceError

SharedDataBaseInfo = json-object
SessionId = json-quotation-mark "SessionId" json-quotation-mark json-name-separator
(json-string | json-null)
SelectCommand = json-quotation-mark "SelectCommand" json-quotation-mark json-name-
separator (json-string | json-null)
ParameterValues = json-quotation-mark "ParameterValues" json-quotation-mark json-
name-separator json-begin-array [ParameterValue *( json-value-separator
ParameterValue) ] json-end-array
OriginalCommand = json-quotation-mark "OriginalCommand" json-quotation-mark json-
name-separator (json-string | json-null)
AggregateExpressions = json-quotation-mark "AggregateExpressions" json-quotation-
mark json-name-separator (json-string | json-null)
AggregateFormatInfos = json-quotation-mark "AggregateFormatInfos" json-quotation-
mark json-name-separator json-begin-array [FormatInfo *( json-value-separator
FormatInfo) ] json-end-array
Restriction = json-quotation-mark "Restriction" json-quotation-mark json-name-
separator (json-string | json-null)
Ordering = json-quotation-mark "Ordering" json-quotation-mark json-name-separator
(json-string | json-null)
AllowEdits = json-quotation-mark "AllowEdits" json-quotation-mark json-name-
separator (json-string | json-null)
AllowAdditions = json-quotation-mark "AllowAdditions" json-quotation-mark json-
name-separator (json-string | json-null)

```

```

AllowDeletions = json-quotation-mark "AllowDeletions" json-quotation-mark json-
name-separator (json-string | json-null)

DataEntry = json-quotation-mark "DataEntry" json-quotation-mark json-name-separator
json-bool
FieldNames = json-quotation-mark "FieldNames" json-quotation-mark json-name-
separator (json-string | json-null)
FormatInfos = json-quotation-mark "FormatInfos" json-quotation-mark json-name-
separator json-begin-array [FormatInfo *( json-value-separator FormatInfo) ] json-
end-array
DataLevelFormat = json-quotation-mark "DataLevelFormat" json-quotation-mark json-
name-separator json-bool
ShowHeaders = json-quotation-mark "ShowHeaders" json-quotation-mark json-name-
separator json-bool
InitialPage = json-quotation-mark "InitialPage" json-quotation-mark json-name-
separator (json-string | json-null)
FetchSchema = json-quotation-mark "FetchSchema" json-quotation-mark json-name-
separator json-bool
FetchKeyFields = json-quotation-mark "FetchKeyFields" json-quotation-mark json-
name-separator json-bool
FetchDisplayFields = json-quotation-mark "FetchDisplayFields" json-quotation-mark
json-name-separator json-bool
DoNotPrefetchImages = json-quotation-mark "DoNotPrefetchImages" json-quotation-mark
json-name-separator json-bool
AutoSumFields = json-quotation-mark "AutoSumFields" json-quotation-mark json-name-
separator JsonArrayOfString
AutoSumFunctions = json-quotation-mark "AutoSumFunctions" json-quotation-mark json-
name-separator JsonArrayOfString

UpdateRecord = json-object
OriginalValues = json-quotation-mark "OriginalValues" json-quotation-mark json-
name-separator JsonArrayOfArrayOfAnyType
NewValues = json-quotation-mark "NewValues" json-quotation-mark json-name-separator
JsonArrayOfArrayOfString
Paging = json-quotation-mark "Paging" json-quotation-mark json-name-separator
PagingInfo
ReturnDataMacroIds = json-quotation-mark "ReturnDataMacroIds" json-quotation-mark
json-name-separator json-bool

CacheCommands = json-int

ClientMessageID = json-string

MessageSeverity = json-string

GetDataRequest = dataBaseInfo-member pagingInfo-member
dataBaseInfo-member = json-quotation-mark "dataBaseInfo" json-quotation-mark json-
name-separator SharedDataBaseInfo
pagingInfo-member = json-quotation-mark "pagingInfo" json-quotation-mark json-name-
separator PagingInfo

GetDataResult = json-null | serviceResult-member
serviceResult-member = json-quotation-mark "d" json-quotation-mark json-name-
separator ServiceResult

InsertRecordsRequest = dataBaseInfo-member updateRecord-member
dataBaseInfo-member = json-quotation-mark "dataBaseInfo" json-quotation-mark json-
name-separator SharedDataBaseInfo

InsertRecordsResult = json-null | serviceResult-member

```

serviceResult-member = json-quotation-mark "d" json-quotation-mark json-name-separator ServiceResult

UpdateRecordsRequest = dataBaseInfo-member updateRecord-member
dataBaseInfo-member = json-quotation-mark "dataBaseInfo" json-quotation-mark json-name-separator SharedDataBaseInfo
updateRecord-member = json-quotation-mark "updateRecord" json-quotation-mark json-name-separator UpdateRecord

UpdateRecordsResult = json-null | serviceResult-member
serviceResult-member = json-quotation-mark "d" json-quotation-mark json-name-separator ServiceResult

DeleteRecordsRequest = dataBaseInfo-member updateRecord-member
dataBaseInfo-member = json-quotation-mark "dataBaseInfo" json-quotation-mark json-name-separator SharedDataBaseInfo
updateRecord-member = json-quotation-mark "updateRecord" json-quotation-mark json-name-separator UpdateRecord

DeleteRecordsResult = json-null | serviceResult-member
serviceResult-member = json-quotation-mark "d" json-quotation-mark json-name-separator ServiceResult

GetDistinctValuesRequest = dataBaseInfo-member columnName-member pagingInfo-member
dataBaseInfo-member = json-quotation-mark "dataBaseInfo" json-quotation-mark json-name-separator SharedDataBaseInfo
columnName-member = json-quotation-mark "columnName" json-quotation-mark json-name-separator json-string
pagingInfo-member = json-quotation-mark "pagingInfo" json-quotation-mark json-name-separator PagingInfo

GetDistinctValuesResult = json-null | serviceResult-member
serviceResult-member = json-quotation-mark "d" json-quotation-mark json-name-separator ServiceResult

FixupRowRequest = dataBaseInfo-member fixupRecord-member
dataBaseInfo-member = json-quotation-mark "dataBaseInfo" json-quotation-mark json-name-separator SharedDataBaseInfo
fixupRecord-member = json-quotation-mark "fixupRecord" json-quotation-mark json-name-separator FixupRecord

FixupRowResult = json-null | serviceResult-member
serviceResult-member = json-quotation-mark "d" json-quotation-mark json-name-separator ServiceResult

GetSearchDataRequest = dataBaseInfo-member
dataBaseInfo-member = json-quotation-mark "dataBaseInfo" json-quotation-mark json-name-separator SharedDataBaseInfo

GetSearchDataResult = json-null | serviceResult-member
serviceResult-member = json-quotation-mark "d" json-quotation-mark json-name-separator ServiceResult

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
- Microsoft Access 2016
- Microsoft SharePoint Server 2016

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

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

8 Change Tracking

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

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

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

- A document revision that incorporates changes to interoperability requirements or functionality.
- The removal of a document from the documentation set.

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

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

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

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

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.
- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- Obsolete document removed.

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

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

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

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

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
Z Appendix B: Product Behavior	Updated list of supported products.	Y	Content updated due to protocol revision.
7 Appendix B: Product Behavior	Updated list of supported products.	Y	Content updated due to protocol revision.

9 Index

A

Abstract data model
[server](#) 27
[Applicability](#) 7

C

[Capability negotiation](#) 8
[Change tracking](#) 58
[Complex types](#) 10

D

Data model – abstract
[server](#) 27
[DeleteRecords service operation example](#) 47

E

Events
timer – server
[session timeout](#) 40

Examples

[DeleteRecords service operation](#) 47
[DeleteRecords Service Operation example](#) 47
[GetData service operation](#) 41
[GetData Service Operation example](#) 41
[InsertRecords service operation](#) 43
[InsertRecords Service Operation example](#) 43
[UpdateRecords service operation](#) 45
[UpdateRecords Service Operation example](#) 45

F

[Fields - vendor-extensible](#) 8

G

[GetData service operation example](#) 41
[Glossary](#) 5

I

[Implementer - security considerations](#) 50
[Index of security parameters](#) 50
[Informative references](#) 6
[InsertRecords service operation example](#) 43
[Introduction](#) 5

M

Message processing
[server](#) 27
Messages
[complex types](#) 10
[simple types](#) 22
[syntax](#) 9
[transport](#) 9

N

[Normative references](#) 6

O

[Overview \(synopsis\)](#) 7

P

[Parameters - security index](#) 50
[Preconditions](#) 7
[Prerequisites](#) 7
[Product behavior](#) 57
Protocol Details
[Server](#) 27
Protocol examples
[DeleteRecords Service Operation](#) 47
[GetData Service Operation](#) 41
[InsertRecords Service Operation](#) 43
[UpdateRecords Service Operation](#) 45

R

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

S

Security
[implementer considerations](#) 50
[parameter index](#) 50
Server
[Abstract data model](#) 27
[Higher-layer triggered events](#) 27
[Initialization](#) 27
[message processing](#) 27
[Message processing events and sequencing rules](#) 27
[Other local events](#) 40
[sequencing rules](#) 27
timer events
[session timeout](#) 40
[Timers](#) 27
[Session timeout](#) 40
[Simple types](#) 22
[Standards assignments](#) 8
[Syntax](#) 9

T

Timer events
server
[session timeout](#) 40
Timers
[server](#) 27
[Tracking changes](#) 58
[Transport](#) 9
Types

[complex](#) 10
[simple](#) 22

U

[UpdateRecords service operation example](#) 45

V

[Vendor-extensible fields](#) 8

[Versioning](#) 8