

[MS-SPSSDBSOGP]: SharePoint Shared Service Database Scale Out Generic Protocol Specification

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.
- **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.

Preliminary Documentation. This Open Specification provides documentation for past and current releases and/or for the pre-release (beta) version of this technology. This Open Specification is final documentation for past or current releases as specifically noted in the document, as applicable; it is preliminary documentation for the pre-release (beta) versions. Microsoft will release final documentation in connection with the commercial release of the updated or new version of this technology. As the documentation may change between this preliminary version and the final version of this technology, there are risks in relying on preliminary documentation. To the extent that you incur additional development obligations or any other costs as a result of relying on this preliminary documentation, you do so at your own risk.

Revision Summary

Date	Revision History	Revision Class	Comments
01/20/2012	0.1	New	Released new document.
04/11/2012	0.1	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	0.1	No change	No changes to the meaning, language, or formatting of the technical content.

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	6
1.4 Relationship to Other Protocols	7
1.5 Prerequisites/Preconditions	7
1.6 Applicability Statement	7
1.7 Versioning and Capability Negotiation	7
1.8 Vendor-Extensible Fields	7
1.9 Standards Assignments	7
2 Messages	8
2.1 Transport	8
2.2 Common Data Types	8
2.2.1 Simple Data Types and Enumerations	8
2.2.1.1 ScaleOutPartitionWeight	8
2.2.1.2 CorrelationId	8
2.2.1.3 Count	8
2.2.1.4 DataRangePoint	8
2.2.1.5 DataSubRangeMode	8
2.2.1.6 IsUpper	8
2.2.1.7 MajorActionType	9
2.2.1.8 MinorActionType	9
2.2.1.9 Minutes	9
2.2.1.10 ScaleOutActionTime	9
2.2.1.11 ScaleOutDatabaseId	9
2.2.1.12 SqlObjectIdentifier	9
2.2.1.13 TotalScaleOutPartitionWeight	9
2.2.2 Bit Fields and Flag Structures	10
2.2.3 Binary Structures	10
2.2.4 Result Sets	10
2.2.4.1 Data Move Plan	10
2.2.4.2 Data Ranges	10
2.2.4.3 Scale-Out Partition Properties	11
2.2.4.4 Scale-Out Data Insert Properties	11
2.2.4.5 Scale-Out Log Entries	11
2.2.5 Tables and Views	13
2.2.6 XML Structures	13
2.2.6.1 Namespaces	13
2.2.6.2 Simple Types	13
2.2.6.3 Complex Types	13
2.2.6.4 Elements	13
2.2.6.5 Attributes	13
2.2.6.6 Groups	13
2.2.6.7 Attribute Groups	13
3 Protocol Details	14
3.1 Common Details	14

3.2	Server Details	14
3.2.1	Abstract Data Model	14
3.2.2	Timers	15
3.2.3	Initialization	15
3.2.4	Higher-Layer Triggered Events	15
3.2.5	Message Processing Events and Sequencing Rules	16
3.2.5.1	proc_ClearDeletedSubRange	16
3.2.5.2	proc_ClearScaleOutLog	17
3.2.5.3	proc_CreateDataMovePlan	18
3.2.5.4	proc_CreateDataRange	19
3.2.5.5	proc_ExtendRange	19
3.2.5.6	proc_GetDataRange	21
3.2.5.7	proc_GetPartitionsCountAndWeight	22
3.2.5.8	proc_MarkDataSubRange	22
3.2.5.9	proc_QueryScaleOutLog	25
3.2.5.10	proc_QueryScaleOutLogWithCorrelationId	25
3.2.5.11	proc_QueryScaleOutLogWithMajorAction	26
3.2.5.12	proc_QueryScaleOutLogWithRangeLimitPoint	26
3.2.5.13	proc_ReadDataSubRange	27
3.2.5.14	proc_RenewScaleOutDatabaseId	29
3.2.6	Timer Events	29
3.2.7	Other Local Events	29
3.3	Client Details	29
3.3.1	Abstract Data Model	30
3.3.2	Timers	30
3.3.3	Initialization	30
3.3.4	Higher-Layer Triggered Events	30
3.3.5	Message Processing Events and Sequencing Rules	30
3.3.6	Timer Events	30
3.3.7	Other Local Events	30
4	Protocol Examples	31
4.1	Data Move between Two Instances of the Same Protocol Server Implementation	31
5	Security	33
5.1	Security Considerations for Implementers	33
5.2	Index of Security Parameters	33
6	Appendix A: Product Behavior	34
7	Change Tracking	35
8	Index	36

1 Introduction

The SharePoint Shared Service Database Scale-Out Generic Protocol specifies the communication sequences used by a protocol client to perform data query and update operations on a protocol server in relation to data ranges and partitions.

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

1.1 Glossary

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

Augmented Backus-Naur Form (ABNF)
Coordinated Universal Time (UTC)

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

data range
result set
return code
Transact-Structured Query Language (T-SQL)

The following terms are specific to this document:

data move: The operation of moving data between two instances of the same implementation of this protocol.

data move chunk: A unit of data, moved during a move operation, that contains a set of scale-out partitions.

data sub-range: A subset of consecutive scale-out partition keys inside a data range. A data-sub range either starts with the same scale-out partition key or ends with the same scale-out partition key as the data range it belongs to does.

data sub-range mode: The runtime state of the data sub-range, which determines the data access conditions for all of the scale-out partitions that have scale-out partition keys falling inside the data sub-range.

inconsistency recovery: The process of recovering from a failed data move operation.

lower data sub-range: A data sub-range that starts with the same scale-out partition key as the data range it belongs to. Only one lower data sub-range can exist inside a data range.

major scale-out action: The primary action that is performed by the protocol client while interacting with the protocol server over a course of time. The primary action can be either a data move or an inconsistency recovery operation.

minor scale-out action: A subaction that occurs while a major scale-out action is executed. Minor scale-out actions can include creating a data sub-range, extending a data range, changing the data sub-range mode of an existing data sub-range, or deleting a data sub-range.

partition data insert stored procedure: A stored procedure that is used to insert a part or all of the data associated with a scale-out partition into the protocol server.

partition data insert table type: A type which represents the schema of a part of the scale-out partition data schema of the protocol implementation.

scale-out log entry: The information that is recorded about a minor scale-out operation after the operation completes.

scale-out partition: A collection of data about a logical entity.

scale-out partition key: A binary value that uniquely identifies a logical entity.

upper data sub-range: A data sub-range that ends with the same scale-out partition key as the data range it belongs to. Only one upper data sub-range can exist inside a data range.

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

1.2 References

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

1.2.1 Normative References

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

[Iseminger] Microsoft Corporation, "SQL Server 2000 Architecture and XML/Internet Support", Volume 1 of Microsoft SQL Server 2000 Reference Library, Microsoft Press, 2001, ISBN 0-7356-1280-3, <http://www.microsoft.com/mspress/books/5001.aspx>

[MSDN-TSQL-Ref] Microsoft Corporation, "Transact-SQL Reference", [http://msdn.microsoft.com/en-us/library/ms189826\(SQL.90\).aspx](http://msdn.microsoft.com/en-us/library/ms189826(SQL.90).aspx)

[MS-TDS] Microsoft Corporation, "[Tabular Data Stream Protocol Specification](#)".

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <http://www.rfc-editor.org/rfc/rfc2119.txt>

[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

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

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

1.3 Overview

This protocol specifies the communications between a database management middle-tier and the database server. The communication satisfies the requests that manage the **data range** and **scale-**

out partitions, for which the protocol server is responsible. The protocol client can send requests to create and remove **data sub-ranges** and can also expand the **data range** on the protocol server. These values determine the way that the protocol server responds to queries made to the **scale-out partitions**. The protocol client can send requests to read or write data into **scale-out partitions** on the protocol server by using this protocol. This mechanism can be used to move **scale-out partitions** data between multiple instances of the same implementation of this protocol.

1.4 Relationship to Other Protocols

The SharePoint Shared Service Database Scale Out Generic Protocol uses the transport stack that is shown in the following diagram.

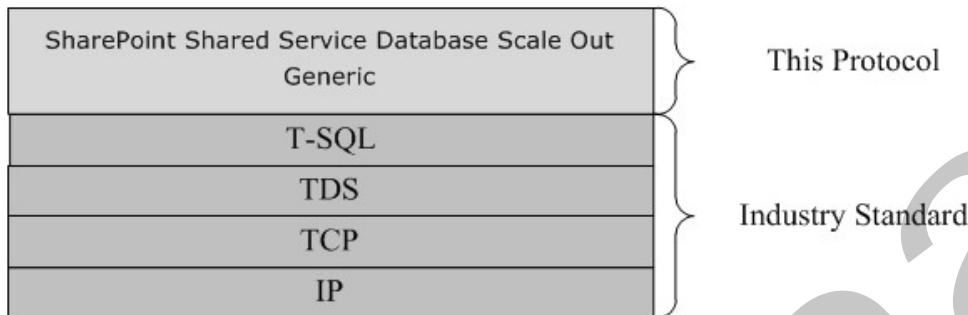


Figure 1: This protocol in relation to other protocols

The implementation of this protocol will specify the value **PartitionKeySize**. This value is used as part of the protocol definition through-out this document.

1.5 Prerequisites/Preconditions

This protocol operates between a protocol client and a protocol server on which the back-end databases are stored. The protocol client is expected to know the location and connection information for the databases.

This protocol requires that the protocol client has the appropriate permissions to call the stored procedures in the back-end databases.

1.6 Applicability Statement

This protocol is intended for use by protocol clients and protocol servers that are connected by high-bandwidth, low-latency network connections.

1.7 Versioning and Capability Negotiation

Security and Authentication Methods: This protocol supports the SSPI and SQL Authentication with the protocol server role, as defined in [\[MS-TDS\]](#).

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

This protocol uses a transport protocol, as specified in [\[MS-TDS\]](#), to call the stored procedures and to return **return codes** and **result sets**.

2.2 Common Data Types

This section defines the common data types that are used by this protocol.

2.2.1 Simple Data Types and Enumerations

2.2.1.1 ScaleOutPartitionWeight

ScaleOutPartitionWeight: int NOT NULL. A number which represents the **weight** attribute of a scale-out partition.

2.2.1.2 CorrelationId

CorrelationId: uniqueidentifier NOT NULL. Identifies an occurrence of a **major scale-out action**.

2.2.1.3 Count

Count: int NOT NULL. A number that specifies the count of records.

2.2.1.4 DataRangePoint

DataRangePoint: varbinary(**PartitionKeySize**) NULL. Identifies a value point in the range of values. NULL is the maximum value.

2.2.1.5 DataSubRangeMode

DataSubRangeMode: tinyint NULL. A number that specifies a **data sub-range mode**. The value MUST be in the following table.

Value	Description
NULL	Represents the case where there is no data sub-range.
1	Represents the data sub-range mode read-only . The partitions which are inside a data sub-range with this mode can be read, but cannot be modified.
2	Represents the data sub-range mode changing . The partitions which are inside a data sub-range with this mode cannot be read or modified.
3	Represents the data sub-range mode deleted . The partitions which are inside a data sub-range with this mode cannot be read or modified.

2.2.1.6 IsUpper

IsUpper: bit NOT NULL. A bit which identifies the lower and the upper part of a data range.

2.2.1.7 MajorActionType

MajorActionType: tinyint NOT NULL. A number that specifies a major scale-out action. The value MUST be in the following table.

Value	Description
0	Represents a data move major scale-out action .
1	Represents an inconsistency recovery major scale-out action .

2.2.1.8 MinorActionType

MinorActionType: tinyint NOT NULL. A number that specifies a **minor scale-out action**. The value MUST be in the following table.

Value	Description
0	Represents the action that a data sub-range is removed.
1	Represents the action that a data sub-range mode is changed into <i>read-only</i> .
2	Represents the action that a data sub-range mode is changed into <i>changing</i> .
3	Represents the action that a data sub-range mode is changed into <i>deleted</i> .
4	Represents the action that a data range is extended.
5	Represents the action that a data sub-range with a data sub-range mode <i>deleted</i> is removed along with all scale-out partitions which have scale-out partition keys falling into that data sub-range .

2.2.1.9 Minutes

Minutes: int NOT NULL. A number which represents a time-span in minutes.

2.2.1.10 ScaleOutActionTime

ScaleOutActionTime: datetime NOT NULL. The **UTC** representation of the time a minor scale-out action starts or ends.

2.2.1.11 ScaleOutDatabaseId

ScaleOutDatabaseId: uniqueidentifier NOT NULL. Identifies the protocol server among a set of protocol servers.

2.2.1.12 SqlObjectIdentifier

SqlObjectIdentifier: nvarchar(128) NOT NULL. The name of a stored procedure or a table type.

2.2.1.13 TotalScaleOutPartitionWeight

TotalScaleOutPartitionWeight: bigint NOT NULL. A number which represents the sum of **weight** attributes of a set of scale-out partitions.

2.2.2 Bit Fields and Flag Structures

No common bit fields or flag structures are defined in this protocol.

2.2.3 Binary Structures

No common binary structures are defined in this protocol.

2.2.4 Result Sets

This section defines the common result sets that are used by this protocol.

The definitions of some of the result sets use **Augmented Backus-Naur Form (ABNF)**, as specified in [\[RFC5234\]](#).

2.2.4.1 Data Move Plan

Data Move Plan result set contains an ordered set of **data move chunk** start or end points.

```
CompositePartitionKey varbinary(max),
```

CompositePartitionKey: A value which represents the start or end point of a **data move chunk**. This value MUST be a **DataRangePoint**, as specified in section [2.2.1.4](#).

2.2.4.2 Data Ranges

Data Ranges result set contains information about the data ranges. Each row in the result set contains all the attributes of a single **data range**.

The **T-SQL** syntax for the result set is as follows:

```
ScaleOutDatabaseId uniqueidentifier,  
RangeStart varbinary(529),  
RangeEnd varbinary(529),  
LowerSubRangePoint varbinary(529),  
LowerSubRangeMode tinyint,  
UpperSubRangePoint varbinary(529),  
UpperSubRangeMode tinyint,
```

ScaleOutDatabaseId: The identifier for the protocol server. The value MUST be a **ScaleOutDatabaseId**, as specified in section [2.2.1.11](#).

RangeStart: The start point of the **data range**. The value MUST be a **DataRangePoint**, as specified in section [2.2.1.4](#).

RangeEnd: The end point of the **data range**. The value MUST be a **DataRangePoint**, as specified in section [2.2.1.4](#).

LowerSubRangePoint: The end point of the **lower data sub-range**. If **LowerSubRangeMode** is NULL, this value MUST be NULL. If **LowerSubRangeMode** is not NULL, this value MUST be greater than or equal to **RangeStart** and less than or equal to **RangeEnd**. If **LowerSubRangeMode** and **UpperSubRangePoint** are both not NULL, this value MUST be less than or equal to **UpperSubRangePoint**. This value MUST be a **DataRangePoint**, as specified in section [2.2.1.4](#).

LowerSubRangeMode: The data sub-range mode of the **lower data sub-range**. If this value is NULL, there is no **lower data sub-range**. This value MUST be a **DataSubRangeMode**, as specified in section [2.2.1.5](#).

UpperSubRangePoint: The start point of the **upper data sub-range**. If **UpperSubRangeMode** is NULL, this value MUST be NULL. If **UpperSubRangeMode** is not NULL, this value MUST be greater than or equal to **RangeStart** and less than or equal to **RangeEnd**. If **LowerSubRangeMode** and **UpperSubRangeMode** are both not NULL, this value MUST be greater than or equal to **LowerSubRangePoint**. This value MUST be a **DataRangePoint**, as specified in section [2.2.1.4](#).

UpperSubRangeMode: The **data sub-range mode** of the **upper data sub-range**. If this value is NULL, there is no **upper data sub-range**. This value MUST be a **DataSubRangeMode**, as specified in section [2.2.1.5](#).

2.2.4.3 Scale-Out Partition Properties

Scale-Out Partition Properties result set contains the total number of scale-out partitions and their total value of the **weight** attributes in the protocol server.

```
Count int,  
TotalWeight bigint,
```

Count: Total number of **scale-out partitions** stored in the protocol server. This value MUST be a **Count**, as specified in section [2.2.1.3](#).

TotalWeight: The sum of **scale-out partition weight** attributes in the protocol server. This value MUST be a **TotalScaleOutPartitionWeight**, as specified in section [2.2.1.13](#).

2.2.4.4 Scale-Out Data Insert Properties

Scale-Out Data Insert Properties result set contains information about the stored procedures and table types to be used when creating scale-out partition data into the protocol server.

The T-SQL syntax for the result set is as follows:

```
Tvp nvarchar(128),  
StoredProcedure nvarchar(128),
```

Tvp: The name of a **partition data insert table type**. This value MUST be a **SqlObjectIdentifier**, as specified in section [2.2.1.12](#).

StoredProcedure: The name of a **partition data insert stored procedure**. This value MUST be a **SqlObjectIdentifier**, as specified in section [2.2.1.12](#).

2.2.4.5 Scale-Out Log Entries

Scale-Out Log Entries result set contains information about the **scale-out log entries**. Each row in the result set contains all the attributes of a single **scale-out log entry**.

The T-SQL syntax for the result set is as follows:

```
MinorActionType tinyint,  
MajorActionType tinyint,
```

CorrelationId uniqueidentifier,
 SubRangePoint varbinary(529),
 RangeLimitPoint varbinary(529),
 TimeStarted datetime,
 Details nvarchar(max),
 TimeCompleted datetime,

MinorActionType: The type of the minor scale-out action that this **scale-out log entry** represents. This value MUST be a **MinorActionType**, as specified in section [2.2.1.8](#).

MajorActionType: The type of the major scale-out action that this **scale-out log entry** represents. This value MUST be a **MajorActionType**, as specified in section [2.2.1.7](#).

CorrelationId: The identifier for the **major scale-out action** instance that the **minor scale-out action**, which this **scale-out log entry** represents, is a part of. This value MUST be a **CorrelationId**, as specified in section [2.2.1.2](#).

SubRangePoint: The **sub-range point** attribute of this **scale-out log entry**. Its value depends on the **MinorActionType** value as specified in the following table.

MinorActionType Value	SubRangePoint Description
0, 1, 2, 3 or 5	If a lower data sub-range is targeted by the minor scale-out action , SubRangePoint represents the end point of that lower data sub-range . If an upper data sub-range is targeted by the minor scale-out action , SubRangePoint represents the start point of that upper data sub-range .
4	If the start point of the data range is extended by the minor scale-out action , SubRangePoint represents the old start point of the data range. If the end point of the data range is extended by the minor scale-out action , SubRangePoint represents the old end point of the data range .

This value MUST be **DataRangePoint**, as specified in section [2.2.1.4](#).

RangeLimitPoint: The **range limit point** attribute of this scale-out log entry of this **scale-out log entry**. Its value depends on the **MinorActionType** value as specified in the following table.

MinorActionType Value	RangeLimitPoint Description
0, 1, 2, 3 or 5	If a lower data sub-range is targeted by the minor scale-out action , RangeLimitPoint represents the start point of that lower data sub-range . If an upper data sub-range is targeted by the minor scale-out action , RangeLimitPoint represents the end point of that upper data sub-range .
4	If the start point of the data range is extended by the minor scale-out action , RangeLimitPoint represents the new start point of the data range . If the end point of the data range is extended by the minor scale-out action , RangeLimitPoint represents the new end point of the data range .

This value MUST be **DataRangePoint**, as specified in section [2.2.1.4](#).

TimeStarted: The time when the **minor scale-out action** that this **scale-out log entry** represents started. This value MUST be a **ScaleOutActionTime**, as specified in section [2.2.1.10](#).

Details: Any information which was stored by the protocol client while initiating the **minor scale-out action** that this **scale-out log entry** represents.

TimeCompleted: The time when the **minor scale-out action** that this **scale-out log entry** represents finished. This value MUST be a **ScaleOutActionTime**, as specified in section [2.2.1.10](#).

2.2.5 Tables and Views

No common table or view structures are defined in this protocol.

2.2.6 XML Structures

No common XML structures are defined in this protocol.

2.2.6.1 Namespaces

This specification does not define any common XML schema namespaces.

2.2.6.2 Simple Types

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

2.2.6.3 Complex Types

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

2.2.6.4 Elements

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

2.2.6.5 Attributes

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

2.2.6.6 Groups

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

2.2.6.7 Attribute Groups

This specification does not define any common XML schema attribute group definitions.

3 Protocol Details

3.1 Common Details

3.2 Server Details

The back-end database server responds only to stored procedure calls from the protocol client. This protocol server returns result sets and return codes and never initiates communication with other endpoints of the protocol.

3.2.1 Abstract Data Model

This section describes a conceptual model of possible data organization which an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

The back-end database server maintains the following sets of data for this protocol:

protocol server identifier: A value that identifies the protocol server instance. This value is of type **ScaleOutDatabaseId**, as specified in section [2.2.1.11](#).

data range: A data range that the protocol server is responsible for.

scale-out partitions: A set of scale-out partitions.

scale-out log entries: A set of scale-out log entries

The **data range** has the following elements:

start point: The start point of the data range. The value is of type **DataRangePoint**, as specified in section [2.2.1.4](#).

end point: The end point of the data range. The value is of type **DataRangePoint**, as specified in section [2.2.1.4](#).

lower data sub-range: A lower data sub-range that is defined on the **data range**. Zero or one occurrences of this sub-range can exist on the **data range**.

upper data sub-range: An upper data sub-range that is defined on the data range. Zero or one occurrences of this sub-range can exist on the **data range**.

The **lower data sub-range** has the following elements:

start point: The start point of the **lower data sub-range**. The value is always the same as that of the start point of the **data range** that the **lower data-sub-range** belongs to. This value is of type **DataRangePoint**, as specified in section [2.2.1.4](#).

end point: The end point of the **lower data sub-range**. The value is of type **DataRangePoint**, as specified in section [2.2.1.4](#).

The **upper data sub-range** has the following elements:

start point: The start point of the **upper data sub-range**. The value is always the same as the start point of the **data range** that the **upper data-sub-range** belongs to. This value is of type **DataRangePoint**, as specified in section [2.2.1.4](#).

end point: The end point of the **upper data sub-range**. The value is of type **DataRangePoint**, as specified in section [2.2.1.4](#).

Each **scale-out partition** has the following elements:

scale-out partition key: The scale-out partition key of the **scale-out partition**. The value is of type **DataRangePoint**, as specified in section [2.2.1.4](#).

scale-out partition weight: The weight, which is based on the data size, of the **scale-out partition**. The value is of type **ScaleOutPartitionWeight**, as specified in section [2.2.1.1](#).

scale-out partition data: Any amount of data, in any data format or type, that is associated with the **scale-out partition**.

Each **scale-out log entry** has the following elements:

start time: The start time of the **scale-out log entry**. The value is of type **ScaleOutActionTime**, as specified in section [2.2.1.10](#).

time completed: The end time of the **scale-out log entry**. The value is of type **ScaleOutActionTime**, as specified in section [2.2.1.10](#).

sub-range point: The sub-range point of the **scale-out log entry**. The value is of type **DataRangePoint**, as specified in section [2.2.1.4](#).

range limit point: The range limit point of **scale-out log entry**. The value is of type **DataRangePoint**, as specified in section [2.2.1.4](#).

major scale-out action: The major scale-out action of the **scale-out log entry**. The value is of type **CorrelationId**, as specified in section [2.2.1.2](#).

major scale-out action type: The **major scale-out action** type of the **scale-out log entry**. The value is of type **MajorActionType**, as specified in section [2.2.1.7](#).

minor scale-out action: The minor scale-out action type of the **scale-out log entry**. The value is of type **MinorActionType**, as specified in section [2.2.1.8](#).

details: The details of the **scale-out log entry**. These details consist of any extra information about the **scale-out log entry**.

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Higher-Layer Triggered Events

None.

3.2.5 Message Processing Events and Sequencing Rules

3.2.5.1 proc_ClearDeletedSubRange

The **proc_ClearDeletedSubRange** stored procedure is called to remove both a data sub-range for which the data sub-range mode is *deleted* and all the scale-out partitions that have scale-out partition keys inside that data sub-range. This stored procedure also creates a scale-out log entry with the specified major scale-out action attributes; the specified log details; and a minor scale-out action type of 5, as specified in section [2.2.1.8](#).

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ClearDeletedSubRange (  
    @Upper bit  
    ,@InitialDeletedSubRangePoint varbinary(529)  
    ,@InitialRangeStart varbinary(529)  
    ,@InitialRangeEnd varbinary(529)  
    ,@MajorActionType tinyint  
    ,@CorrelationId uniqueidentifier  
    ,@LogDetails nvarchar(max)  
    ,@ErrorCode int OUTPUT  
);
```

@Upper: A value that specifies whether the data sub-range to be removed is the lower data sub-range or the upper data sub-range. This value MUST be of type **IsUpper**, as specified in section [2.2.1.6](#).

@InitialDeletedSubRangePoint: The start or end point of the data sub-range to be removed. If this data sub-range is a lower data sub-range, the value represents the end point; otherwise, the value represents the start point. This value also represents the **sub-range point** attribute of the scale-out log entry to be created. The value MUST be of type **DataRangePoint**, as specified in section [2.2.1.4](#).

@InitialRangeStart: The start point of the data range. If the data sub-range to be removed is the lower data sub-range, this value also represents the **range limit point** attribute of the scale-out log entry to be created. The value MUST be of type **DataRangePoint**, as specified in section [2.2.1.4](#).

@InitialRangeEnd: The end point of the data range. If the data sub-range to be removed is the upper data sub-range, this value also represents the **range limit point** attribute of the scale-out log entry to be created. The value MUST be of type **DataRangePoint**, as specified in section [2.2.1.4](#).

@MajorActionType: The type of the major scale-out action of the scale-out log entry to be created. The value MUST be of type **MajorActionType**, as specified in section [2.2.1.7](#).

@CorrelationId: The identifier of the major scale-out action of the scale-out log entry to be created. The value MUST be of type **CorrelationId**, as specified in section [2.2.1.2](#).

@LogDetails: Any custom information to be stored as part of the scale-out log entry to be created.

@ErrorCode: The error code. On return from this stored procedure, the value MUST be set to one of the integers that is listed in the following table.

Value	Description
-3	<p>One of the following errors occurred:</p> <ul style="list-style-type: none"> ▪ The start point of the data range is not the same as the value of <i>@InitialRangeStart</i>. ▪ The end point of the data range is not the same as the value of <i>@InitialRangeEnd</i>. ▪ The value of <i>@Upper</i> is 0, and the value of <i>@InitialDeletedSubRangePoint</i> is not the same as the end point of the lower data sub-range. ▪ The value of <i>@Upper</i> is 0, and the data sub-range mode of the lower data sub-range is not <i>deleted</i>. ▪ The value of <i>@Upper</i> is 1, and the value of <i>@InitialDeletedSubRangePoint</i> is not the same as the upper data sub-range start point. ▪ The value of <i>@Upper</i> is 1, and the data sub-range mode of the upper data sub-range is not <i>deleted</i>.
0	No errors occurred.

Return Values: This stored procedure returns an integer that MUST be 0.

Result Sets: This stored procedure MUST NOT return any result sets.

3.2.5.2 `proc_ClearScaleOutLog`

The `proc_ClearScaleOutLog` stored procedure is called to remove all the scale-out log entries if no scale-out log entry exists such that the difference between the value of its **time completed** attribute and the current time is less than the specified time-out value.

The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_ClearScaleOutLog (
  @LogEntryTimeout int
  ,@ErrorCode int OUTPUT
);

```

@LogEntryTimeout: The scale-out log entry time-out value. The value MUST be of type **Minutes**, as specified in section [2.2.1.9](#).

@ErrorCode: The error code. On return from this stored procedure, the value MUST be set to one of the integers that is listed in the following table.

Value	Description
-11	The difference between the value of the time completed attribute of a scale-out log entry and the current time is less than the value of <i>@LogEntryTimeout</i> .
0	No errors occurred.

Return Values: This stored procedure returns an integer that MUST be 0.

Result Sets: This stored procedure MUST NOT return any result sets.

3.2.5.3 proc_CreateDataMovePlan

The **proc_CreateDataMovePlan** stored procedure is called to get a set of data move chunk start or end points in either descending or ascending order, respectively. If start points are returned, the end point of the first data move chunk is the end point of the data range, and the end point of each data move chunk, except for the first, is the same as the start point of the previous data move chunk. If end points are returned, the start point of the first data move chunk is the start point of the data range, and the start point of each data move chunk, except for the first, is the same as the end point of the previous data move chunk. Start points **MUST** be returned in descending order, and end points **MUST** be returned in ascending order.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_CreateDataMovePlan (  
    @Upper bit  
    ,@ChunkSize int  
    ,@WeightToMove bigint  
    ,@TotalWeight bigint  
);
```

@Upper: A value that specifies whether to return data move chunk start or end points. If the value is 0, end points **MUST** be returned. If the value is 1, start points **MUST** be returned. The value **MUST** be of type **IsUpper**, as specified in section [2.2.1.6](#).

@ChunkSize: The limit for the sum of **weight** attributes of the scale-out partitions in a data move chunk. The sum of the **weight** attributes of the scale-out partitions that have scale-out partition keys falling into each data move chunk **MUST NOT** exceed this limit by more than half of the **weight** attribute for the scale-out partition that has the closest scale-out partition key to the limit point of that data move chunk. The sum of **weight** attributes of the scale-out partitions that have scale-out partition keys falling into each data move chunk **MUST** be as close as possible to this value. If the value of *@Upper* is 0, the end point of a data move chunk is the limit point of that data move chunk. If the value of *@Upper* is 1, the start point of a data move chunk is the limit point of that data move chunk. The value **MUST** be of type **TotalScaleOutPartitionWeight**, as specified in section [2.2.1.1](#).

@WeightToMove: The limit for the sum of **weight** attributes of all the scale-out partitions that have scale-out partition keys falling into any of the data move chunks that are represented by the data move chunk limit points to be returned. The sum of the **weight** attributes of the scale-out partitions in all the data move chunks that have the returned data move chunk limit points **MUST NOT** exceed this value by more than half of the **weight** attribute for the scale-out partition that has the closest scale-out partition key to the last data move chunk limit point returned. The sum of weight attributes of the scale-out partitions in all the data move chunks with the returned data move chunk limit points **MUST** be as close as possible to this value. If the value of *@Upper* is 0, the end point of a data move chunk is the limit point of that data move chunk. If the value of *@Upper* is 1, the start point of a data move chunk is the limit point of that data move chunk. The value **MUST** be of type **TotalScaleOutPartitionWeight**, as specified in section [2.2.1.13](#).

@TotalWeight: The sum of **weight** attributes of all the scale-out partitions that are stored on the protocol server. The value **MUST** be of type **TotalScaleOutPartitionWeight**, as specified in section [2.2.1.13](#).

Return Values: This stored procedure returns an integer that **MUST** be 0.

Result Sets:

This stored procedure **MUST** return a [Data Move Plan](#)

3.2.5.4 proc_CreateDataRange

The **proc_CreateDataRange** stored procedure is called to create a data range with the specified start and end points. The data range is assigned a random identifier at this time.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_CreateDataRange (  
    @RangeStart varbinary(max)  
    ,@RangeEnd varbinary(max)  
    ,@ErrorCode int OUTPUT  
);
```

@RangeStart: The start point of the data range to be created. The value MUST be of type **DataRangePoint**, as specified in section [2.2.1.4](#).

@RangeEnd: The end point of the data range to be created. The value MUST be of type **DataRangePoint**, as specified in section [2.2.1.4](#).

@ErrorCode: The error code. On return from this stored procedure, the value MUST be set to one of the integers that is listed in the following table.

Value	Description
-1	The data range already exists.
0	No errors occurred.

Return Values: This stored procedure returns an integer that MUST be 0.

Result Sets: This stored procedure MUST NOT return any result sets.

3.2.5.5 proc_ExtendRange

The **proc_ExtendRange** stored procedure is called to extend a data range at either its start or its end point. If specified, this stored procedure will also create a data sub-range with a data sub-range mode of *changing* for the whole extension. If the data range is extended by modifying its start point, the new data sub-range will have the new start point of the data range as its start point and the old start point of the data range as its end point. If the data range is extended by modifying its end point, the new data sub-range will have the new end point of the data range as its end point and the old end point of the data range as its start point. This stored procedure also creates a new scale-out log entry with the specified major scale-out action properties, the specified log details, and a minor scale-out action type of 4, as specified in section [2.2.1.8](#).

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ExtendRange (  
    @RangePoint varbinary(max)  
    ,@Upper bit  
    ,@AsChanging bit  
    ,@InitialRangeStart varbinary(max)  
    ,@InitialRangeEnd varbinary(max)  
    ,@InitialSubRangePoint varbinary(max)  
    ,@InitialSubRangeMode tinyint  
    ,@InitialOppositeSubRangePoint varbinary(max)
```

```

,@InitialOppositeSubRangeMode tinyint
,@MajorActionType tinyint
,@CorrelationId uniqueidentifier
,@LogDetails nvarchar(max)
,@ErrorCode int OUTPUT
);

```

@RangePoint: The new start or end point of the data range. This value is also the **range limit point** attribute of the scale-out log entry to be created. The value MUST be of type **DataRangePoint**, as specified in section [2.2.1.4](#).

@Upper: A value that specifies whether the start or the end point will be modified to extend the data range. If the value is 0, the start point will be modified; otherwise, the end point will be modified. The value MUST be of type **IsUpper**, as specified in section [2.2.1.6](#).

@AsChanging: A value that specifies whether a data sub-range with a data sub-range mode of *changing* will be created for the extension.

@InitialRangeStart: The start point of the data range. If the value of *@Upper* is 0, this value is also the **sub-range point** attribute of the scale-out log entry to be created. The value MUST be of type **DataRangePoint**, as specified in section [2.2.1.4](#).

@InitialRangeEnd: The end point of the data range. If the value of *@Upper* is 1, this value is also the **sub-range point** attribute of the scale-out log entry to be created. The value MUST be of type **DataRangePoint**, as specified in section [2.2.1.4](#).

@InitialSubRangePoint: The start or end point of the lower data sub-range or upper data sub-range. If the value of *@Upper* is 0, this value represents the end point of the lower data sub-range; otherwise, this value represents the start point of the upper data sub-range. This value MUST be of type **DataRangePoint**, as specified in section [2.2.1.4](#).

@InitialSubRangeMode: The data sub-range mode of the lower data sub-range or upper data sub-range. If the value of *@Upper* is 0, this value represents the data sub-range mode of the lower data sub-range; otherwise, this value represents the data sub-range mode of the upper data sub-range. This value MUST be of type **DataSubRangeMode**, as specified in section [2.2.1.5](#).

@InitialOppositeSubRangePoint: The start or end point of the lower data sub-range or upper data sub-range. If the value of *@Upper* is 0, this value represents the start point of the upper data sub-range; otherwise, this value represents the end point of the lower data sub-range. This value MUST be of type **DataRangePoint**, as specified in section [2.2.1.4](#).

@InitialOppositeSubRangeMode: The data sub-range mode of the lower data sub-range or upper data sub-range. If the value of *@Upper* is 0, this value represents the data sub-range mode of the upper data sub-range; otherwise, this value represents the data sub-range mode of the lower data sub-range. This value MUST be of type **DataSubRangeMode**, as specified in section [2.2.1.5](#).

@MajorActionType: The type of the major scale-out action of the scale-out log entry to be created. The value MUST be of type **MajorActionType**, as specified in section [2.2.1.7](#).

@CorrelationId: The identifier of the major scale-out action of the scale-out log entry to be created. The value MUST be of type **CorrelationId**, as specified in section [2.2.1.2](#).

@LogDetails: Any custom information to be stored as part of the scale-out log entry to be created.

@ErrorCode: The error code. On return from this stored procedure, the value MUST be set to one of the integers that is listed in the following table.

Value	Description
-6	<p>One of the following errors occurred:</p> <ul style="list-style-type: none"> ▪ The value of <i>@InitialSubRangeMode</i> is not NULL, and the value of <i>@AsChanging</i> is 1. ▪ The value of <i>@InitialSubRangeMode</i> is neither NULL nor 1.
-5	<p>One of the following errors occurred:</p> <ul style="list-style-type: none"> ▪ The value of <i>@Upper</i> is 0, and the value of <i>@RangePoint</i> is greater than or equal to the value of <i>@InitialRangeStart</i>. ▪ The value of <i>@Upper</i> is 1, and the value of <i>@RangePoint</i> is less than or equal to the value of <i>@InitialRangeEnd</i>.
-3	<p>One of the following errors occurred:</p> <ul style="list-style-type: none"> ▪ The start point of the data range is not the same as the value of <i>@InitialRangeStart</i>. ▪ The end point of the data range is not the same as the value of <i>@InitialRangeEnd</i>. ▪ The value of <i>@Upper</i> is 0, and the value of <i>@InitialSubRangePoint</i> is not the same as the end point of the lower data sub-range. ▪ The value of <i>@Upper</i> is 0, and the value of <i>@InitialSubRangeMode</i> is not the same as the data sub-range mode of the lower data sub-range. ▪ The value of <i>@Upper</i> is 0, and the value of <i>@InitialOppositeSubRangePoint</i> is not the same as the start point of upper data sub-range. ▪ The value of <i>@Upper</i> is 0, and the value of <i>@InitialOppositeSubRangeMode</i> is not the same as the data sub-range mode of the upper data sub-range. ▪ The value of <i>@Upper</i> value is 1, and the value of <i>@InitialSubRangePoint</i> is not the same as the start point of the upper data sub-range. ▪ The value of <i>@Upper</i> is 1, and the value of <i>@InitialSubRangeMode</i> is not the same as the data sub-range mode of the upper data sub-range. ▪ The value of <i>@Upper</i> is 1, and the value of <i>@InitialOppositeSubRangePoint</i> is not the same as the end point of the lower data sub-range. ▪ The value of <i>@Upper</i> is 1, and the value of <i>@InitialOppositeSubRangeMode</i> is not the same as the data sub-range mode of the lower data sub-range.
0	No errors occurred.

Return Values: This stored procedure returns an integer that MUST be 0.

Result Sets: This stored procedure MUST NOT return any result sets.

3.2.5.6 **proc_GetDataRange**

The **proc_GetDataRange** stored procedure is called to get the data range.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetDataRange (
);
```

Return Values: This stored procedure returns an integer that MUST be 0.

Result Sets:

This stored procedure MUST return a [Data Ranges](#)

3.2.5.7 proc_GetPartitionsCountAndWeight

The **proc_GetPartitionsCountAndWeight** stored procedure is called to get the total scale-out partition count and the sum of **weight** attributes of all the scale-out partitions that are stored on the protocol server.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetPartitionsCountAndWeight (
);
```

Return Values: This stored procedure returns an integer that MUST be 0.

Result Sets:

This stored procedure MUST return a [Scale-Out Partition Properties](#)

3.2.5.8 proc_MarkDataSubRange

The **proc_MarkDataSubRange** stored procedure is called to create a new data sub-range or modify an existing one. If a lower data sub-range is targeted and one already exists, the data sub-range mode and end point of the lower data sub-range are updated; otherwise, a lower data sub-range is created with the specified data sub-range mode and end point. If an upper data sub-range is targeted and one already exists, the data sub-range mode and start point of the upper data sub-range are updated; otherwise, an upper data sub-range is created with the specified data sub-range mode and start point. This stored procedure also creates a scale-out log entry with the specified major scale-out action attributes and the specified log details.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_MarkDataSubRange (
  @SubRangePoint varbinary(max)
  ,@SubRangeMode tinyint
  ,@Upper bit
  ,@InitialRangeStart varbinary(max)
  ,@InitialRangeEnd varbinary(max)
  ,@InitialSubRangePoint varbinary(max)
  ,@InitialSubRangeMode tinyint
  ,@InitialOppositeSubRangePoint varbinary(max)
  ,@InitialOppositeSubRangeMode tinyint
  ,@MajorActionType tinyint
  ,@CorrelationId uniqueidentifier
  ,@LogDetails nvarchar(max)
  ,@ErrorCode int OUTPUT
);
```

@SubRangePoint: The new start or end point for the specified data sub-range. If a lower data sub-range is targeted, this value represents the new end point; otherwise, this value represents the new start point. The value also represents the **sub-range point** attribute of the scale-out log entry to create. This value MUST be of type **DataRangePoint**, as specified in section [2.2.1.4](#).

@SubRangeMode: The new data sub-range mode for the specified data sub-range. The minor scale-out action of the scale-out log entry depends on this value. The correlation between these two values is listed in the following table.

Value of @SubRangeMode	Minor Scale-Out Action
NULL	0
1	1
2	2
3	3

The value MUST be of type **DataSubRangeMode**, as specified in section [2.2.1.5](#).

@Upper: A value that specifies whether the targeted data sub-range is the lower data sub-range or the upper data sub-range. This value MUST be of type **IsUpper**, as specified in section [2.2.1.6](#).

@InitialRangeStart: The start point of the data range. If the targeted data sub-range is the lower data sub-range, this value also represents the **range limit point** attribute of the scale-out log entry to create. The value MUST be of type **DataRangePoint**, as specified in section [2.2.1.4](#).

@InitialRangeEnd: The end point of the data range. If the targeted data sub-range is the upper data sub-range, this value also represents the **range limit point** attribute of the scale-out log entry to create. The value MUST be of type **DataRangePoint**, as specified in section [2.2.1.4](#).

@InitialSubRangePoint: The current start or end point for the targeted data sub-range. If the targeted data sub-range is the lower data sub-range, this value represents the current end point of the targeted data sub-range; otherwise, this value represents the current start point. If the targeted data sub-range does not exist, this value MUST be NULL. The value MUST be of type **DataRangePoint**, as specified in section [2.2.1.4](#).

@InitialSubRangeMode: The current data sub-range mode of the targeted data sub-range. If the targeted data sub-range does not exist, this value MUST be NULL. The value MUST be of type **DataSubRangeMode**, as specified in section [2.2.1.5](#).

@InitialOppositeSubRangePoint: The current start or end point of the opposite data sub-range. (If the targeted data sub-range is the lower data sub-range, the opposite data sub-range is the upper data sub-range. If the targeted data sub-range is the upper data sub-range, the opposite data sub-range is the lower data sub-range.) If the opposite data sub-range is the lower data sub-range, this value represents the current end point of the lower data sub-range; otherwise, this value represents the current start point of the upper data sub-range. If the opposite data sub-range does not exist, this value MUST be NULL. The value MUST be of type **DataRangePoint**, as specified in section [2.2.1.4](#).

@InitialOppositeSubRangeMode: The current data sub-range mode of the opposite data sub-range. If the opposite data sub-range does not exist, this value MUST be NULL. (If the targeted data sub-range is the lower data sub-range, the opposite data sub-range is the upper data sub-range. If the targeted data sub-range is the upper data sub-range, the opposite data sub-range is the lower data sub-range.) The value MUST be of type **DataSubRangeMode**, as specified in section [2.2.1.5](#).

@MajorActionType: The type of the major scale-out action of the scale-out log entry to be created. The value MUST be of type **MajorActionType**, as specified in section [2.2.1.7](#).

@CorrelationId: The identifier of the major scale-out action of the scale-out log entry to be created. The value MUST be of type **CorrelationId**, as specified in section [2.2.1.2](#).

@LogDetails: Any custom information to be stored as part of the scale-out log entry to be created.

@ErrorCode: The error code. On return from this stored procedure, the value MUST be set to one of the integers that is listed in the following table.

Value	Description
-10	One of the following errors occurred: <ul style="list-style-type: none"> The value of <i>@Upper</i> is 0, the value of <i>@InitialOppositeSubRangeMode</i> is not NULL, the value of <i>@SubRangeMode</i> is not NULL and the value of <i>@SubRangePoint</i> is greater than the value of <i>@InitialOppositeSubRangePoint</i>. The value of <i>@Upper</i> is 1, the value of <i>@InitialOppositeSubRangeMode</i> is not NULL, the value of <i>@SubRangeMode</i> is not NULL, and the value of <i>@SubRangePoint</i> is less than the value of <i>@InitialOppositeSubRangePoint</i>.
-9	The value of <i>@InitialSubRangeMode</i> is 2, and the value of <i>@SubRangeMode</i> is NULL.
-8	The value of <i>@InitialSubRangeMode</i> is 1, and the value of <i>@SubRangeMode</i> value is 2.
-7	One of the following errors occurred: <ul style="list-style-type: none"> The value of <i>@Upper</i> is 0, the value of <i>@InitialSubRangeMode</i> is not NULL, and the value of <i>@SubRangePoint</i> is less than the value of <i>@InitialSubRangePoint</i>. The value of <i>@Upper</i> is 1, the value of <i>@InitialSubRangeMode</i> is not NULL, and the value of <i>@SubRangePoint</i> is greater than the value of <i>@InitialSubRangePoint</i>.
-4	The value of <i>@InitialSubRangeMode</i> is 3, and the value of <i>@SubRangeMode</i> is not 3.
-3	One of the following errors occurred: <ul style="list-style-type: none"> The start point of the data range is not the same as the value of <i>@InitialRangeStart</i>. The end point of the data range is not the same as the value of <i>@InitialRangeEnd</i>. The value of <i>@Upper</i> is 0, and the value of <i>@InitialSubRangePoint</i> is not the same as the end point of the lower data sub-range. The value of <i>@Upper</i> is 0, and the value of <i>@InitialSubRangeMode</i> is not the same as the data sub-range mode of the lower data sub-range. The value of <i>@Upper</i> is 0, and the value of <i>@InitialOppositeSubRangePoint</i> is not the same as the start point of the upper data sub-range. The value of <i>@Upper</i> is 0, and the value of <i>@InitialOppositeSubRangeMode</i> is not the same as the data sub-range mode of the upper data sub-range. The value of <i>@Upper</i> is 1, and the value of <i>@InitialSubRangePoint</i> is not the same as the start point of the upper data sub-range. The value of <i>@Upper</i> is 1, and the value of <i>@InitialSubRangeMode</i> is not the same as the data

Value	Description
	<p>sub-range mode of the upper data sub-range.</p> <ul style="list-style-type: none"> ▪ The value of <i>@Upper</i> is 1, and the value of <i>@InitialOppositeSubRangePoint</i> is not the same as the end point of the lower data sub-range. ▪ The value of <i>@Upper</i> is 1, and the value of <i>@InitialOppositeSubRangeMode</i> is not the same as the data sub-range mode of the lower data sub-range.
-2	<p>One of the following errors occurred:</p> <ul style="list-style-type: none"> ▪ The value of <i>@Upper</i> is 0, the value of <i>@InitialSubRangeMode</i> is not NULL, and the value of <i>@SubRangePoint</i> is either greater than the value of <i>@InitialRangeEnd</i> or less than or equal to the value of <i>@InitialRangeStart</i>. ▪ The value of <i>@Upper</i> is 1, the value of <i>@InitialSubRangeMode</i> is not NULL, and the value of <i>@SubRangePoint</i> is either less than the value of <i>@InitialRangeStart</i> or greater than or equal to the value of <i>@InitialRangeEnd</i>.
0	No errors occurred.

Return Values: This stored procedure returns an integer that MUST be 0.

Result Sets: This stored procedure MUST NOT return any result sets.

3.2.5.9 **proc_QueryScaleOutLog**

The **proc_QueryScaleOutLog** stored procedure is called to get the specified number of scale-out log entries that have the most recent **time completed** attribute.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_QueryScaleOutLog (
    @Count int
);
```

@Count: The number of scale-out log entries to be returned. The value MUST be of type **Count**, as specified in section [2.2.1.3](#).

Return Values: This stored procedure returns an integer that MUST be 0.

Result Sets:

This stored procedure MUST return a [Scale-Out Log Entries](#)

3.2.5.10 **proc_QueryScaleOutLogWithCorrelationId**

The **proc_QueryScaleOutLogWithCorrelationId** stored procedure is called to get the specified number of scale-out log entries that have the most recent **time completed** attribute, that have the specified major scale-out action type, and that have the specified major scale-out action identifier.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_QueryScaleOutLogWithCorrelationId (
    @MajorActionType tinyint
```

```
,@CorrelationId uniqueidentifier
,@Count int
);
```

@MajorActionType: The major scale-out action type of the scale-out log entries to be returned. The value MUST be of type **MajorActionType**, as specified in section [2.2.1.7](#).

@CorrelationId: The major scale-out action identifier of the scale-out log entries to be returned. The value MUST be of type **CorrelationId**, as specified in section [2.2.1.2](#).

@Count: The number of scale-out log entries to be returned. The value MUST be of type **Count**, as specified in section [2.2.1.3](#).

Return Values: This stored procedure returns an integer that MUST be 0.

Result Sets:

This stored procedure MUST return a [Scale-Out Log Entries](#)

3.2.5.11 **proc_QueryScaleOutLogWithMajorAction**

The **proc_QueryScaleOutLogWithMajorAction** stored procedure is called to get the specified number of scale-out log entries that have both the most recent **time completed** attribute and the specified major scale-out action type.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_QueryScaleOutLogWithMajorAction (
@MajorActionType tinyint
,@Count int
);
```

@MajorActionType: The major scale-out action type of the scale-out log entries to be returned. The value MUST be of type **MajorActionType**, as specified in section [2.2.1.7](#).

@Count: The number of scale-out log entries to be returned. The value MUST be of type **Count**, as specified in section [2.2.1.3](#).

Return Values: This stored procedure returns an integer that MUST be 0.

Result Sets:

This stored procedure MUST return a [Scale-Out Log Entries](#)

3.2.5.12 **proc_QueryScaleOutLogWithRangeLimitPoint**

The **proc_QueryScaleOutLogWithRangeLimitPoint** stored procedure is called to get the specified number of scale-out log entries that have the most recent **time completed** attribute, that have the specified **range limit point** attribute, that have the specified major scale-out action type, and that have the specified major scale-out action identifier.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_QueryScaleOutLogWithRangeLimitPoint (
@MajorActionType tinyint
```

```

, @CorrelationId uniqueidentifier
, @RangeLimitPoint varbinary(max)
, @Count int
);

```

@MajorActionType: The major scale-out action type of the scale-out log entries to be returned. The value MUST be of type **MajorActionType**, as specified in section [2.2.1.7](#).

@CorrelationId: The major scale-out action identifier of the scale-out log entries to be returned. The value MUST be of type **CorrelationId**, as specified in section [2.2.1.2](#).

@RangeLimitPoint: The **range limit point** attribute of the scale-out log entries to be returned. The value MUST be of type **DataRangePoint**, as specified in section [2.2.1.4](#).

@Count: The number of scale-out log entries to be returned. The value MUST be of type **Count**, as specified in section [2.2.1.3](#).

Return Values: This stored procedure returns an integer that MUST be 0.

Result Sets:

This stored procedure MUST return a [Scale-Out Log Entries](#)

3.2.5.13 proc_ReadDataSubRange

The **proc_ReadDataSubRange** stored procedure is called to get a list of partition data insert stored procedure-partition data insert table type pairs and a list of result sets that contain data for a set of scale-out partitions.

The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_ReadDataSubRange (
@Upper bit
, @ReadSubRangePoint varbinary(529)
, @InitialRangeStart varbinary(529)
, @InitialRangeEnd varbinary(529)
, @ErrorCode int OUTPUT
);

```

@Upper: A value that specifies whether the data sub-range from which to read data is a lower data sub-range or an upper data sub-range. Note that the partition data result sets, as specified later in this section, return data for the scale-out partitions that have scale-out partition keys inside that data sub-range. The value MUST be of type **IsUpper**, as specified in section [2.2.1.6](#).

@ReadSubRangePoint: The start or end point of the data sub-range from which to read data. If the data sub-range is a lower data sub-range, this value represents the end point of the data sub-range; otherwise, this value represents the start point. Note that the partition data result sets, as specified later in this section, return data for the scale-out partitions that have scale-out partition keys inside that data sub-range. The value MUST be of type **DataRangePoint**, as specified in section [2.2.1.4](#).

@InitialRangeStart: The start point of the data range. The value MUST be of type **DataRangePoint**, as specified in section [2.2.1.4](#).

@InitialRangeEnd: The end point of the data range. The value MUST be of type **DataRangePoint**, as specified in section [2.2.1.4](#).

@ErrorCode: The error code. On return from this stored procedure, the value MUST be set to one of the integers that is listed in the following table.

Value	Description
-3	The data range does not have the start or end point that is specified by the <i>@InitialRangeStart</i> or <i>@InitialRangeEnd</i> parameter, respectively, or a data sub-range does not exist that has both a data sub-range mode of <i>read-only</i> and the values that are specified by the <i>@Upper</i> and <i>@ReadSubRangePoint</i> parameters.
0	No errors occurred.

Return Values: This stored procedure returns an integer that MUST be 0.

Result Sets:

This stored procedure MUST return a [Scale-Out Data Insert Properties](#) as its first result set. This stored procedure MUST return at least one partition data result set. A partition data result set is a result set that contains any data for a partition for which the schema and content vary among protocol implementations. Each partition data result set can have any number of columns that have any data type. The number of partition data result sets returned from an implementation of this stored procedure MUST be the same as the number of rows returned by the **Scale-Out Data Insert Properties** (section [2.2.4.4](#)) result set. The data contained in any partition data result set MUST belong to a scale-out partition that has a scale-out partition key inside the data sub-range specified by the *@Upper* and *@ReadSubRangePoint* parameters. A partition data insert table type that has the name specified by the **Tvp** attribute in any row of the first returned result set MUST exist in the protocol implementation. A partition data insert table type that has the name specified by the **Tvp** attribute in the first row of the first returned result set MUST have exactly the same column names and data types as the first partition data result set. The same correspondence MUST exist for the second row and second partition data result set, the third row and third partition data result set, and so on.

A partition data insert stored procedure that has the name specified by the **StoredProcedure** attribute in any row of the first result set MUST exist in the protocol implementation, and it MUST have the following functionality and schema.

Functionality: This stored procedure stores the data of a set of scale-out partitions.

Schema: This stored procedure MUST have the following parameters.

```
@Data <partition data insert table type> READONLY,  
@Upper bit,  
@ChangingSubRangePoint varbinary(PartitionKeySize),  
@InitialRangeStart varbinary(PartitionKeySize),  
@InitialRangeEnd varbinary(PartitionKeySize),  
@ErrorCode int OUTPUT
```

@Data: The data to be stored for the scale-out partitions. The scale-out partition keys of the scale-out partitions for which this parameter has data MUST exist in the data sub-range that is specified by the *@Upper* and *@ChangingSubRangePoint* parameters. Note that <partition data insert table type> MUST be the partition data insert table type that has the name specified by the **Tvp** attribute in the same row that the name of this stored procedure is specified in the **Scale-Out Data Insert Properties** (section [2.2.4.4](#)) result set.

@Upper: A value that specifies whether the data sub-range is a lower data sub-range or upper data sub-range. The value MUST be of type **IsUpper**, as specified in section [2.2.1.6](#).

@ChangingSubRangePoint: The start or end point of the data sub-range. If the data sub-range is a lower data sub-range, this value represents the end point of the data sub-range; otherwise, this value represents the start point. The value MUST be of type **DataRangePoint**, as specified in section [2.2.1.4](#).

@InitialRangeStart: The start point of the data range. The value MUST be of type **DataRangePoint**, as specified in section [2.2.1.4](#).

@InitialRangeEnd: The end point of the data range. The value MUST be of type **DataRangePoint**, as specified in section [2.2.1.4](#).

@ErrorCode: The error code. On return from this stored procedure, the value MUST be set to one of the integers that is listed in the following table.

Value	Description
-3	The data range does not have the start or end point that is specified by the <i>@InitialRangeStart</i> or <i>@InitialRangeEnd</i> parameter, respectively, or a data sub-range does not exist that has both a data sub-range mode of <i>changing</i> and the values that are specified by the <i>@Upper</i> and <i>@ChangingSubRangePoint</i> parameters.
0	No errors occurred.

3.2.5.14 **proc_RenewScaleOutDatabaseId**

The **proc_RenewScaleOutDatabaseId** stored procedure is called to assign a new random identifier to the data range.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_RenewScaleOutDatabaseId (  
);
```

Return Values: This stored procedure returns an integer that MUST be 0.

Result Sets: This stored procedure MUST NOT return any result sets.

3.2.6 **Timer Events**

None.

3.2.7 **Other Local Events**

None.

3.3 **Client Details**

The protocol client acts as a client when it calls the back-end database server, requesting the processing of stored procedures and optionally caching some of the data that is retrieved by the stored procedures.

3.3.1 Abstract Data Model

None.

3.3.2 Timers

None.

3.3.3 Initialization

None.

3.3.4 Higher-Layer Triggered Events

None.

3.3.5 Message Processing Events and Sequencing Rules

The protocol client handles each stored procedure with the same basic processing method of calling the stored procedure and waiting for the return code and any result sets to be returned.

3.3.6 Timer Events

None.

3.3.7 Other Local Events

None.

4 Protocol Examples

This section provides an example of the communication between a protocol client and a protocol server. The example describes the details of how this protocol can be used to move data between two instances of the same implementation of the protocol server.

4.1 Data Move between Two Instances of the Same Protocol Server Implementation

This example illustrates how a database management middle-tier can move scale-out partitions from one instance of the protocol server to another having the same implementation. To facilitate reading ease and comprehension, the term protocol "Source Server" means the protocol server "Source Server," and the term protocol "Target Server" means the protocol server "Target Server."

The following steps are executed:

1. The user requests the protocol client to move the **scale-out partitions** which have a total weight equal to 50% of the total weight of all **scale-out partitions** on the protocol "Source Server" to the protocol "Target Server."
2. The protocol client calls **proc_GetPartitionsCountAndWeight** on the protocol "Source Server."
3. The protocol "Source Server" responds with the total count and the total weight of all **scale-out partitions** it has.
4. The protocol "Source Server" returns a return code that the protocol client ignores.
5. The protocol client calculates the 50% of the total weight and obtains the weight to move.
6. The protocol client calls **proc_CreateDataMovePlan** on the protocol "Source Server" by passing in the total weight and the weight to move calculated in step 5.
7. The protocol "Source Server" responds with a list of scale-out partition keys.
8. The protocol "Source Server" returns a return code that the protocol client ignores.
9. For each **scale-out partition key (SPK)** the following steps are executed:
 1. The protocol client calls **proc_MarkDataSubRange** on the protocol "Source Server" to create a data sub-range bounded by the **SPK** and the start or end point of the current data range of the protocol "Source Server." The protocol client passes in the data sub-range mode as *read-only* to this call.
 2. The protocol "Source Server" creates a **data sub-range** with the specified values and sets *@ErrorCode* to 0.
 3. The protocol "Source Server" returns a return code that the protocol client ignores.
 4. The protocol client calls **proc_ExtendRange** on the protocol "Target Server" to extend the **data range** up to the **SPK** on the protocol "Target Server."
 5. The protocol "Target Server" extends the **data range** accordingly and creates a **data sub-range** covering the whole extension in **data sub-range mode changing**. It sets *@ErrorCode* to 0.
 6. The protocol "Target Server" returns a return code that the protocol client ignores.

7. The protocol client calls **proc_ReadDataSubRange** on the protocol "Source Server" for the **data sub-range** created in step b.
 8. The protocol "Source Server" responds with the a list of partition data insert stored procedure, partition data insert table type pairs and a list of result sets containing data for the scale-out partitions in the specified **data sub-range**. It sets *@ErrorCode* to 0.
 9. The protocol "Source Server" returns a return code that the protocol client ignores.
 10. The protocol client populates the data into the returned **partition data insert table types** with the data coming from the result set, which is at the same index as the **partition data insert table type**, and calls each corresponding **partition data insert stored procedure** with the populated **partition data insert table type** on the protocol "Target Server."
 11. The protocol "Target Server" inserts all the data passed into the **partition data insert stored procedures**. It sets *@ErrorCode* to 0 on each one of the **partition data insert stored procedure** calls.
 12. The protocol "Target Server" returns a return code that the protocol client ignores from each one of the **partition data insert stored procedure** calls.
 13. The protocol client calls **proc_MarkDataSubRange** on the protocol "Target Server" to set the **data sub-range mode** of the extension **data sub-range** to *read-only*.
 14. The protocol "Target Server" sets the **data sub-range mode** on the specified **data sub-range** to *read-only* and sets *@ErrorCode* to 0.
 15. The protocol "Target Server" returns a return code that the protocol client ignores.
 16. The protocol client calls **proc_MarkDataSubRange** on the protocol "Source Server" to set the **data sub-range mode** of the **data sub-range** created in step b to *deleted*.
 17. The protocol "Source Server" sets the **data sub-range mode** on the specified **data sub-range** to *deleted* and sets *@ErrorCode* to 0.
 18. The protocol "Source Server" returns a return code that the protocol client ignores.
 19. The protocol client calls **proc_MarkDataSubRange** on the protocol "Target Server" to set the **data sub-range mode** of the extension **data sub-range** to *read-write*.
 20. The protocol "Target Server" sets the **data sub-range mode** on the specified **data sub-range** to *read-write* and sets *@ErrorCode* to 0.
 21. The protocol "Target Server" returns a return code that the protocol client ignores.
 22. The protocol client calls **proc_ClearDeletedSubRange** on the protocol "Source Server" to delete the **data sub-range** created in step b.
 23. The protocol "Source Server" deletes the specified **data sub-range** along with all the data associated with the **scale-out partitions** inside that **data sub-range** and sets *@ErrorCode* to 0.
 24. The protocol "Source Server" returns a return code that the protocol client ignores.
10. The protocol client notifies the user that the data move operation is finished.

5 Security

5.1 Security Considerations for Implementers

Interactions with SQL are susceptible to tampering and other forms of security risks. Implementers are advised to sanitize input parameters for stored procedures before invoking them.

5.2 Index of Security Parameters

None.

6 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- Microsoft® SharePoint® Foundation 2013 Preview
- Microsoft® SharePoint® Server 2013 Preview

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.

7 Change Tracking

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

Preliminary

8 Index

A

Abstract data model

[client](#) 30

[server](#) 14

[Applicability](#) 7

[Attribute groups - overview](#) 13

[Attributes - overview](#) 13

B

[Binary structures - overview](#) 10

[Bit fields - overview](#) 10

C

[Capability negotiation](#) 7

[Change tracking](#) 35

Client

[abstract data model](#) 30

[higher-layer triggered events](#) 30

[initialization](#) 30

[local events](#) 30

[message processing](#) 30

[overview](#) 29

[sequencing rules](#) 30

[timer events](#) 30

[timers](#) 30

Common data types

[overview](#) 8

[Complex types - overview](#) 13

[CorrelationId simple type](#) 8

[Count simple type](#) 8

D

Data model - abstract

[client](#) 30

[server](#) 14

Data types

[common](#) 8

[CorrelationId simple type](#) 8

[Count simple type](#) 8

[DataRangePoint simple type](#) 8

[DataSubRangeMode simple type](#) 8

[IsUpper simple type](#) 8

[MajorActionType simple type](#) 9

[MinorActionType simple type](#) 9

[Minutes simple type](#) 9

[ScaleOutActionTime simple type](#) 9

[ScaleOutDatabaseId simple type](#) 9

[ScaleOutPartitionWeight simple type](#) 8

[SqlObjectIdentifier simple type](#) 9

[TotalScaleOutPartitionWeight simple type](#) 9

Data types - simple

[CorrelationId](#) 8

[Count](#) 8

[DataRangePoint](#) 8

[DataSubRangeMode](#) 8

[IsUpper](#) 8

[MajorActionType](#) 9

[MinorActionType](#) 9

[Minutes](#) 9

[ScaleOutActionTime](#) 9

[ScaleOutDatabaseId](#) 9

[ScaleOutPartitionWeight](#) 8

[SqlObjectIdentifier](#) 9

[TotalScaleOutPartitionWeight](#) 9

[DataRangePoint simple type](#) 8

[DataSubRangeMode simple type](#) 8

E

[Elements - overview](#) 13

Events

[local - client](#) 30

[local - server](#) 29

[timer - client](#) 30

[timer - server](#) 29

F

[Fields - vendor-extensible](#) 7

[Flag structures - overview](#) 10

G

[Glossary](#) 5

[Groups - overview](#) 13

H

Higher-layer triggered events

[client](#) 30

[server](#) 15

I

[Implementer - security considerations](#) 33

[Index of security parameters](#) 33

[Informative references](#) 6

Initialization

[client](#) 30

[server](#) 15

[Introduction](#) 5

[IsUpper simple type](#) 8

L

Local events

[client](#) 30

[server](#) 29

M

[MajorActionType simple type](#) 9

Message processing

[client](#) 30

Messages

- [attribute groups](#) 13
- [attributes](#) 13
- [binary structures](#) 10
- [bit fields](#) 10
- [common data types](#) 8
- [complex types](#) 13
- [elements](#) 13
- [flag structures](#) 10
- [groups](#) 13
- [namespaces](#) 13
- [result sets](#) 10
- [simple types](#) 13
- [table structures](#) 13
- [transport](#) 8
- [view structures](#) 13
- [XML structures](#) 13

Methods

- [proc_ClearDeletedSubRange](#) 16
- [proc_ClearScaleOutLog](#) 17
- [proc_CreateDataMovePlan](#) 18
- [proc_CreateDataRange](#) 19
- [proc_ExtendRange](#) 19
- [proc_GetDataRange](#) 21
- [proc_GetPartitionsCountAndWeight](#) 22
- [proc_MarkDataSubRange](#) 22
- [proc_QueryScaleOutLog](#) 25
- [proc_QueryScaleOutLogWithCorrelationId](#) 25
- [proc_QueryScaleOutLogWithMajorAction](#) 26
- [proc_QueryScaleOutLogWithRangeLimitPoint](#) 26
- [proc_ReadDataSubRange](#) 27
- [proc_RenewScaleOutDatabaseId](#) 29

- [MinorActionType simple type](#) 9

- [Minutes simple type](#) 9

N

- [Namespaces](#) 13

- [Normative references](#) 6

O

- [Overview \(synopsis\)](#) 6

P

- [Parameters - security index](#) 33

- [Preconditions](#) 7

- [Prerequisites](#) 7

- [proc_ClearDeletedSubRange method](#) 16

- [proc_ClearScaleOutLog method](#) 17

- [proc_CreateDataMovePlan method](#) 18

- [proc_CreateDataRange method](#) 19

- [proc_ExtendRange method](#) 19

- [proc_GetDataRange method](#) 21

- [proc_GetPartitionsCountAndWeight method](#) 22

- [proc_MarkDataSubRange method](#) 22

- [proc_QueryScaleOutLog method](#) 25

- [proc_QueryScaleOutLogWithCorrelationId method](#) 25

- [proc_QueryScaleOutLogWithMajorAction method](#) 26

- [proc_QueryScaleOutLogWithRangeLimitPoint method](#) 26

- [proc_ReadDataSubRange method](#) 27

- [proc_RenewScaleOutDatabaseId method](#) 29

- [Product behavior](#) 34

R

- [References](#) 6

- [informative](#) 6

- [normative](#) 6

- [Relationship to other protocols](#) 7

- [Result sets - overview](#) 10

S

- [ScaleOutActionTime simple type](#) 9

- [ScaleOutDatabaseId simple type](#) 9

- [ScaleOutPartitionWeight simple type](#) 8

Security

- [implementer considerations](#) 33

- [parameter index](#) 33

Sequencing rules

- [client](#) 30

Server

- [abstract data model](#) 14

- [higher-layer triggered events](#) 15

- [initialization](#) 15

- [local events](#) 29

- [overview](#) 14

- [proc_ClearDeletedSubRange method](#) 16

- [proc_ClearScaleOutLog method](#) 17

- [proc_CreateDataMovePlan method](#) 18

- [proc_CreateDataRange method](#) 19

- [proc_ExtendRange method](#) 19

- [proc_GetDataRange method](#) 21

- [proc_GetPartitionsCountAndWeight method](#) 22

- [proc_MarkDataSubRange method](#) 22

- [proc_QueryScaleOutLog method](#) 25

- [proc_QueryScaleOutLogWithCorrelationId method](#) 25

- [proc_QueryScaleOutLogWithMajorAction method](#) 26

- [proc_QueryScaleOutLogWithRangeLimitPoint method](#) 26

- [proc_ReadDataSubRange method](#) 27

- [proc_RenewScaleOutDatabaseId method](#) 29

- [timer events](#) 29

- [timers](#) 15

Simple data types

- [CorrelationId](#) 8

- [Count](#) 8

- [DataRangePoint](#) 8

- [DataSubRangeMode](#) 8

- [IsUpper](#) 8

- [MajorActionType](#) 9

- [MinorActionType](#) 9

- [Minutes](#) 9

- [ScaleOutActionTime](#) 9

- [ScaleOutDatabaseId](#) 9

- [ScaleOutPartitionWeight](#) 8

- [SqlObjectIdentifier](#) 9

[TotalScaleOutPartitionWeight](#) 9
[Simple types - overview](#) 13
[SqlObjectIdentifier simple type](#) 9
[Standards assignments](#) 7
Structures
 [binary](#) 10
 [table and view](#) 13
 [XML](#) 13

T

[Table structures - overview](#) 13
Timer events
 [client](#) 30
 [server](#) 29
Timers
 [client](#) 30
 [server](#) 15
[TotalScaleOutPartitionWeight simple type](#) 9
[Tracking changes](#) 35
[Transport](#) 8
Triggered events - higher-layer
 [client](#) 30
 [server](#) 15
Types
 [complex](#) 13
 [simple](#) 13

V

[Vendor-extensible fields](#) 7
[Versioning](#) 7
[View structures - overview](#) 13

X

[XML structures](#) 13