

# [MS-TSSPROC]: Translation Services Stored Procedures 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](mailto: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

<b>1 Introduction</b>	<b>5</b>
1.1 Glossary	5
1.2 References	5
1.2.1 Normative References	5
1.2.2 Informative References	6
1.3 Overview	6
1.4 Relationship to Other Protocols	6
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
<b>2 Messages</b>	<b>8</b>
2.1 Transport	8
2.2 Common Data Types	8
2.2.1 Simple Data Types and Enumerations	8
2.2.2 Bit Fields and Flag Structures	8
2.2.3 Binary Structures	8
2.2.4 Result Sets	8
2.2.4.1 proc_GetConversionBatch.ResultSet0	8
2.2.4.2 proc_GetConversionBatchInterleaved.ResultSet0	9
2.2.4.3 proc_GetGroups.ResultSet0	9
2.2.4.4 proc_GetItems.ResultSet0	10
2.2.4.5 proc_GetJobs.ResultSet0	10
2.2.4.6 proc_GetJobStatus.ResultSet0	11
2.2.4.7 proc_UpdateConversionBatch.ResultSet0	12
2.2.5 Tables and Views	12
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
2.2.6.4.1 databaseBatchUpdate	13
2.2.6.4.2 databaseJobAdd	15
2.2.6.5 Attributes	16
2.2.6.6 Groups	16
2.2.6.7 Attribute Groups	16
<b>3 Protocol Details</b>	<b>17</b>
3.1 Translation Server Details	17
3.1.1 Abstract Data Model	17
3.1.2 Timers	18
3.1.3 Initialization	18
3.1.4 Higher-Layer Triggered Events	18
3.1.5 Message Processing Events and Sequencing Rules	19
3.1.5.1 proc_AddGroup	20
3.1.5.2 proc_AddJob	20
3.1.5.3 proc_CancelAllActiveJobs	21
3.1.5.4 proc_CancelJob	22

3.1.5.5	proc_GetConversionBatch .....	22
3.1.5.6	proc_GetConversionBatchInterleaved .....	23
3.1.5.7	proc_GetGroups .....	24
3.1.5.8	proc_GetItemCount .....	24
3.1.5.9	proc_GetItems .....	25
3.1.5.10	proc_GetJobs .....	26
3.1.5.11	proc_GetJobStatus .....	27
3.1.5.12	proc_HasActiveJobs .....	27
3.1.5.13	proc_JobsExpire .....	28
3.1.5.14	proc_SubmitJob .....	29
3.1.5.15	proc_UpdateConversionBatch .....	29
3.1.5.16	proc_UpdateFailedItem .....	30
3.1.5.17	proc_UpdateItemCount .....	31
3.1.5.18	proc_UpdateSucceededItem .....	31
3.1.6	Timer Events .....	32
3.1.7	Other Local Events .....	32
3.2	Client Details .....	32
<b>4</b>	<b>Protocol Examples .....</b>	<b>33</b>
<b>5</b>	<b>Security .....</b>	<b>37</b>
5.1	Security Considerations for Implementers .....	37
5.2	Index of Security Parameters .....	37
<b>6</b>	<b>Appendix A: Product Behavior .....</b>	<b>38</b>
<b>7</b>	<b>Change Tracking .....</b>	<b>39</b>
<b>8</b>	<b>Index .....</b>	<b>40</b>

# 1 Introduction

The Translation Services Stored Procedures Protocol allows protocol clients to store and retrieve information about machine translating documents from one language to another.

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 [\[MS-GLOS\]](#):

### Security Support Provider Interface (SSPI)

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

**result set**  
**return code**  
**SQL authentication**  
**stored procedure**  
**Structured Query Language (SQL)**  
**Transact-Structured Query Language (T-SQL)**  
**translation group**  
**translation item**  
**translation job**  
**XML element**  
**XML namespace**  
**XML schema**

The following terms are specific to this document:

**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](mailto: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>

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

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

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

### 1.2.2 Informative References

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

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

### 1.3 Overview

This protocol allows clients to add, modify, and delete **translation jobs**, **translation groups** and **translation items** from a database on the protocol server, as well as retrieve those translation jobs, translation groups and translation items by using predefined criteria.

Section 3.1.1 introduces and defines common terminology used throughout this document.

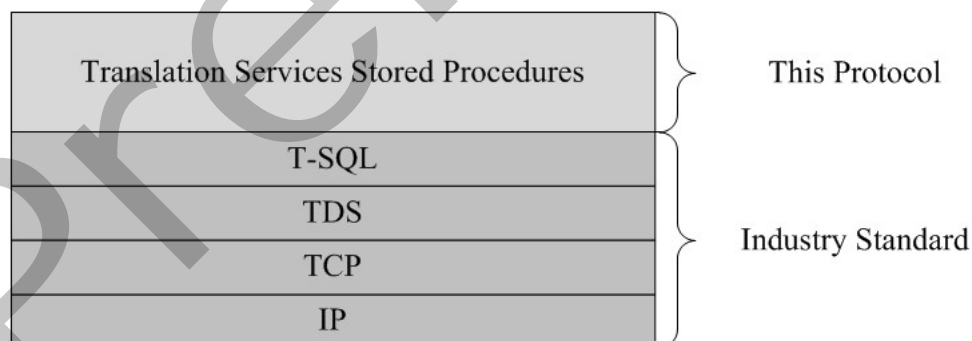
Section 3.1.5 gives an overview of the **stored procedures** in this protocol.

### 1.4 Relationship to Other Protocols

This protocol relies on [\[MS-TDS\]](#) as its transport protocol to call stored procedures to store and interact with translation jobs, translation groups, and translation items. The stored procedures accomplish this through **result sets** and **return codes**.

Database queries or calls to stored procedures and the returned result sets are written in the [\[MSDN-TSQL-Ref\]](#) language.

The following diagram shows the transport stack that the protocol uses:



## Figure 1: This protocol in relation to other protocols

### 1.5 Prerequisites/Preconditions

The operations described by this protocol operate between a protocol client and a protocol server. The client is expected to have the location and connection information for the required databases on the protocol server.

This protocol requires that the protocol client have the appropriate permissions to call the stored procedures in the required databases on the protocol server.

### 1.6 Applicability Statement

This protocol is designed to work only with the translation jobs, translation groups, and translation items specified in this document.

### 1.7 Versioning and Capability Negotiation

- **Supported Transports:** this protocol uses [\[MS-TDS\]](#) as specified in section [2.1](#).
- **Security and Authentication Methods:** this protocol supports the **Security Support Provider Interface (SSPI)** and **SQL authentication** with the protocol server role as described in [MS-TDS].

### 1.8 Vendor-Extensible Fields

None.

### 1.9 Standards Assignments

None.

## 2 Messages

### 2.1 Transport

The Tabular Data Stream Protocol specifies the transport protocol used to call stored procedures, query **SQL** tables, get return codes, and return result sets.

### 2.2 Common Data Types

#### 2.2.1 Simple Data Types and Enumerations

No common simple data types or enumerations are defined in this protocol.

#### 2.2.2 Bit Fields and Flag Structures

No common bit field 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 specifies the result sets that are used for this protocol.

##### 2.2.4.1 `proc_GetConversionBatch.ResultSet0`

The `proc_GetConversionBatch.ResultSet0` result set contains a set of translation items.

```
JobId bigint,  
GroupId smallint,  
ItemId int,  
InProgress bit,  
InputFile nvarchar(max),  
OutputFile nvarchar(max),  
AttemptsRemaining tinyint,  
WorkerServerInstance uniqueidentifier,  
StartTime datetime,  
CreateTime datetime,
```

**JobId:** The job identifier of the translation item.

**GroupId:** The group identifier of the translation item.

**ItemId:** The item identifier of the translation item.

**InProgress:** The status of the transaction item. If this field is zero, the translation item has not started. If this field is one, the translation item is already in progress but is stale. See section [3.1.5.5](#) for more details.

**InputFile:** The input file location of the translation item.

**OutputFile:** The output file location for the translation item.

**AttemptsRemaining:** The number of translation attempts remaining for the translation item.



**WorkerServerInstance:** The unique identifier of the worker server instance that is handling the translation of the translation item. If the **InProgress** attribute of the result set is zero, this field MUST be NULL.

**StartTime:** The time translation started for the translation item. If the **InProgress** attribute of the result set is zero, this field MUST be NULL.

**CreateTime:** The creation time of the translation job.

#### 2.2.4.2 **proc\_GetConversionBatchInterleaved.ResultSet0**

The **proc\_GetConversionInterleavedBatch.ResultSet0** result set contains a set of translation items.

```
JobId bigint,  
GroupId smallint,  
ItemId int,  
InProgress bit,  
InputFile nvarchar(max),  
OutputFile nvarchar(max),  
AttemptsRemaining tinyint,  
WorkerServerInstance uniqueidentifier,  
StartTime datetime,  
CreateTime datetime,
```

**JobId:** The job identifier of the translation item.

**GroupId:** The group identifier of the translation item.

**ItemId:** The item identifier of the translation item.

**InProgress:** The status of the translation item. If this field is zero, the translation item has not started. If this field is one, the translation item is already in progress but is stale. See section [3.1.5.6](#) for more details.

**InputFile:** The input file location of the translation item.

**OutputFile:** The output file location for the translation item.

**AttemptsRemaining:** The number of the translation attempts remaining for the translation item.

**WorkerServerInstance:** The unique identifier of the worker server instance that is handling the translation of the translation item. If the **InProgress** attribute of the result set is zero, this field MUST be NULL.

**StartTime:** The time translation started for the translation item. If the **InProgress** attribute of the result set is zero, this field MUST be NULL.

**CreateTime:** The time the translation job was created.

#### 2.2.4.3 **proc\_GetGroups.ResultSet0**

The **proc\_GetGroups.ResultSet0** result set contains information about translation groups in a translation job. Each row represents a translation group and contains additional information from the translation job.

```
GroupId smallint,  
InputRoot nvarchar(max),  
OutputRoot nvarchar(max),  
CreateTime datetime,  
CancelTime datetime,  
Submitted bit,  
Settings nvarchar(max),
```

**GroupId:** The group identifier of the translation group.

**InputRoot:** The input root of the translation group. See section [3.1.1](#) for more details.

**OutputRoot:** The output root of the translation group. See section [3.1.1](#) for more details.

**CreateTime:** The time the translation job was created.

**CancelTime:** The time the translation job was cancelled.

**Submitted:** A flag indicating whether or not the translation job has been submitted for translation.

**Settings:** The settings of the translation job. See section [3.1.1](#) for more details.

#### 2.2.4.4 `proc_GetItems.ResultSet0`

The **`proc_GetItems.ResultSet0`** result set contains information about translation items in a translation job. Each row represents a translation item.

```
ItemId int,  
StartTime datetime,  
StopTime datetime,  
ErrorCode int,  
InputFile nvarchar(max),  
OutputFile nvarchar(max),
```

**ItemId:** The item identifier of the translation item.

**StartTime:** The time translation started for the translation item.

**StopTime:** The time translation finished for the translation item.

**ErrorCode:** The error code identifier for the translation item. See section [3.1.1](#) for more details.

**InputFile:** The input file location of the translation item.

**OutputFile:** The output file location for the translation item.

#### 2.2.4.5 `proc_GetJobs.ResultSet0`

The **`proc_GetJobs.ResultSet0`** result set contains information about a list of translation jobs . Each row represents a translation job and information about it.

```
JobId bigint,  
CreateTime datetime,  
CancelTime datetime,  
Submitted bit,
```

Name nvarchar(max),

**JobId:** The job identifier of the translation job.

**CreateTime:** The time the translation job was created.

**CancelTime:** The time the translation job was cancelled.

**Submitted:** A flag indicating whether or not the translation job has been submitted for translation.

**Name:** The name associated with the translation job.

#### 2.2.4.6 **proc\_GetJobStatus.ResultSet0**

The **proc\_GetJobStatus.ResultSet0** result set contains information about the translation items in a translation job.

```
Total int,  
NotSubmitted int,  
NotStarted int,  
InProgress int,  
Succeeded int,  
Failed int,  
Canceled int,  
Name nvarchar(max),
```

**Total:** The total number of translation items in the translation job.

**NotSubmitted:** The number of translation items in the translation job that has the following:

- the **Submitted** attribute of the translation job is zero.
- the **CancelTime** attribute of the translation job is NULL.

**NotStarted:** This is the number of translation items in the translation job that has the following:

- the **Submitted** attribute of the translation job is one.
- the **CancelTime** attribute of the translation job is NULL.
- the **StartTime** attribute of the translation item is NULL.

**InProgress:** This is the number of translation items in the translation job that has the following:

- the **Submitted** attribute of the translation job is one.
- the **CancelTime** attribute of the translation job is NULL.
- the **StartTime** attribute of the translation item is not NULL.
- the **StopTime** attribute of the translation item is NULL.

**Succeeded:** This is the number of translation items in the translation job that has the following:

- the **Submitted** attribute of the translation job is one.
- the **StartTime** attribute of the translation item is not NULL.

- the **StopTime** attribute of the translation item is not NULL.
- the **ErrorCode** attribute of the translation item is NULL.

**Failed:** This is the number of translation items in the translation job that has the following:

- the **Submitted** attribute of the translation job is one.
- the **StartTime** attribute of the translation item is not NULL.
- the **StopTime** attribute of the translation item is not NULL.
- the **ErrorCode** attribute of the translation item is not NULL.

**Canceled:** This is the number of translation items in the translation job that has the following:

- the **Submitted** attribute of the translation job is one.
- the **CancelTime** attribute of the translation job is not NULL.
- the **StopTime** attribute of the translation item is NULL.

**Name:** This is the **Name** of the translation job.

#### 2.2.4.7 **proc\_UpdateConversionBatch.ResultSet0**

The **proc\_UpdateConversionBatch.ResultSet0** result set contains information about the translation jobs and translation groups for the translation items that were updated in the **proc\_UpdateConversionBatch** stored procedure (section [3.1.5.15](#)).

```
JobId bigint,
GroupId smallint,
InputRoot nvarchar(max),
OutputRoot nvarchar(max),
Settings nvarchar(max),
PartitionId uniqueidentifier,
UserToken varbinary(max),
```

**JobId:** The job identifier of the translation job.

**GroupId:** The group identifier of the translation group.

**InputRoot:** The input root of the translation group. See section [3.1.1](#) for more details.

**OutputRoot:** The output root of the translation group. See section [3.1.1](#) for more details.

**Settings:** The settings of the translation job. See section [3.1.1](#) for more details.

**PartitionId:** The partition identifier on which the translation job runs.

**UserToken:** The credentials of the user that submitted the translation job.

#### 2.2.5 **Tables and Views**

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

## 2.2.6 XML Structures

The namespaces (section [2.2.6.1](#)), simple types (section [2.2.6.2](#)), complex types (section [2.2.6.3](#)), and elements (section [2.2.6.4](#)) that are specified in this section are used in the **databaseBatchUpdate** (section [2.2.6.4.1](#)) XML element and the **databaseJobAdd** (section [2.2.6.4.2](#)) XML element.

The syntax of the definitions in this section uses XML schema as defined in [\[XMLSCHEMA1\]](#) and [\[XMLSCHEMA2\]](#).

### 2.2.6.1 Namespaces

This specification defines and references various XML namespaces using the mechanisms specified in [\[XMLNS\]](#). Although this specification associates a specific prefix for each XML namespace that is used, the choice of any particular XML namespace prefix is implementation-specific and not significant for interoperability.

Prefix	Namespace URI	Reference
none	<a href="http://schemas.microsoft.com/office/server/translation/2010/11/databaseBatchUpdate">http://schemas.microsoft.com/office/server/translation/2010/11/databaseBatchUpdate</a>	Section <a href="#">2.2.6.4.1</a>
none	<a href="http://schemas.microsoft.com/office/server/translation/2010/11/databaseJobAdd">http://schemas.microsoft.com/office/server/translation/2010/11/databaseJobAdd</a>	Section <a href="#">2.2.6.4.2</a>
xs	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>	<a href="#">[XMLSCHEMA1]</a> ↓ <a href="#">[XMLSCHEMA2]</a> ↓

### 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

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

Element	Description
databaseBatchUpdate	Represents a set of translation items that will be updated in the database.
databaseJobAdd	Represents a translation group that will be added to the database.

#### 2.2.6.4.1 databaseBatchUpdate

This is an XML structure that represents a set of translation items that will be updated in the database.

```

<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/office/server/translation/2010/11/databaseBatch
Update" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.microsoft.com/office/server/translation/2010/11/databaseBatchUpdate">
  <xs:simpleType name="guid">
    <xs:restriction base="xs:string">
      <xs:pattern value="[\da-fA-F]{8}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-
F]{12}" />
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="batch">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="start">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="0" maxOccurs="unbounded" name="item">
                <xs:complexType>
                  <xs:attribute name="job" type="xs:long" use="required" />
                  <xs:attribute name="group" type="xs:short" use="required" />
                  <xs:attribute name="id" type="xs:int" use="required" />
                  <xs:attribute name="wsi" type="guid" use="required" />
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="failed">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="0" maxOccurs="unbounded" name="item">
                <xs:complexType>
                  <xs:attribute name="job" type="xs:long" use="required" />
                  <xs:attribute name="group" type="xs:short" use="required" />
                  <xs:attribute name="id" type="xs:int" use="required" />
                  <xs:attribute name="error" type="xs:int" use="required" />
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

**batch.start.item.job:** A long integer that represents the job identifier of the translation item. This element is used with the **group** and **id** attributes to identify a translation item.

**batch.start.item.group:** A short integer that represents the group identifier of the translation item.

**batch.start.item.id:** An integer that represents the item identifier of the translation item.

**batch.start.item.wsi:** A unique identifier that represents the new value for the worker server instance of the translation item.

**batch.failed.item.job:** A long that represents the job identifier of the translation item. This element is used with the **group** and **id** attributes to identify a translation item.

**batch.failed.item.group:** A short integer that represents the group identifier of the translation item.

**batch.failed.item.id:** An integer that represents the item identifier of the translation item.

**batch.failed.item.error:** An integer that represents the new error code value for the translation item.

Example:

```
<batch
xmlns="http://schemas.microsoft.com/office/server/translation/2010/11/databaseBatchUpdate">
  <start>
    <item job="-6843074718075247457" group="1" id="1" wsi="b00ae9a1-0474-474e-b348-
f6a8bcc95331" />
    <item job="-6843074718075247457" group="1" id="2" wsi="b00ae9a1-0474-474e-b348-
f6a8bcc95331" />
  </start>
  <failed>
    <item job="-6843074718075247457" group="1" id="3" error="10" />
    <item job="-6843074718075247457" group="1" id="4" error="11" />
  </failed>
</batch>
```

#### 2.2.6.4.2 databaseJobAdd

This is an XML structure that represents a translation group that will be added to the database.

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/office/server/translation/2010/11/databaseJobAdd"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="group">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" name="item">
          <xs:complexType>
            <xs:attribute name="id" type="xs:int" use="required" />
            <xs:attribute name="in" type="xs:string" use="required" />
            <xs:attribute name="out" type="xs:string" use="required" />
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

**Group.item.id:** An integer that represents the item identifier of the translation item.

**Group.item.in:** A string that represents the input file location of the translation item.

**Group.item.out:** A string that represents the output file location for the translation item.

Example:

```
<group xmlns="http://schemas.microsoft.com/office/server/translation/2010/11/databaseJobAdd">
  <item id="1" in="Aenean%20nec.docx" out="Aenean%20nec_Spanish.docx" />
  <item id="2" in="Fusce%20aliquet.docx" out="Fusce%20aliquet_Spanish.docx" />
  <item id="3" in="Lorem%20ipsum.docx" out="Lorem%20ipsum_Spanish.docx" />
  <item id="4" in="Nunc%20viverra.docx" out="Nunc%20viverra_Spanish.docx" />
  <item id="5" in="Pellentesque.docx" out="Pellentesque_Spanish.docx" />
</group>
```

### **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 Translation Server Details

#### 3.1.1 Abstract Data Model

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

A translation item is the atomic unit of file translation. A translation item contains information about translating a single file. A translation item is uniquely identified by the **JobId**, **GroupId**, and **ItemId**. The information associated with a translation item is specified as follows:

- **JobId**: A bigint that identifies a translation job to which the translation item belongs. This attribute MUST NOT be NULL.
- **ItemId**: An integer that identifies a translation item within the translation job. Two translation items within a translation group MUST NOT have the same **ItemId**. This attribute MUST NOT be NULL.
- **GroupId**: A smallint that identifies to which translation group in the translation job that this translation item belongs. This attribute MUST NOT be NULL.
- **StartTime**: A datetime that specifies when the protocol client reported that it began translating the file specified by **InputFile**.
- **StopTime**: A datetime that specifies when the protocol client reported that it had finished translating the file specified by **InputFile**.
- **AttemptsRemaining**: A tinyint that specifies the number of remaining attempts for translation the translation item has. The meaning of the values in this field is defined by the protocol client. This attribute MUST NOT be NULL.
- **InputFile**: A string that represents the location of the source file relative to the translation group **InputRoot**. This attribute MUST NOT be NULL.
- **OutputFile**: A string that represents the location of the destination file relative to the translation group **OutputRoot**, indicating where the translated file will be placed. This attribute MUST NOT be NULL.
- **WorkerServerInstance**: A unique identifier that specifies the worker server instance handling the translation of this translation item. The meaning of the values in this field is defined by the protocol client.
- **ErrorCode**: An integer that represents the error message identifier the protocol client encountered. The meaning of the values in this field is defined by the protocol client.

A translation group comprises one or more translation items. A translation group is uniquely identified by a **JobId** and a **GroupId**. The information associated with a translation group is specified as follows:

- **JobId**: A bigint that identifies a translation job to which the translation group belongs. This attribute MUST NOT be NULL.

- **GroupId:** A smallint that identifies the translation group within the translation job. Two translation groups within a translation job MUST NOT have the same **GroupId**. This attribute MUST NOT be NULL.
- **InputRoot:** A nvarchar(max) that specifies common portions of the input file path that all translation items in this translation group share. This attribute MUST NOT be NULL.
- **OutputRoot:** A nvarchar(max) that specifies common portions of the output file path that all translation items in this translation group share. This attribute MUST NOT be NULL.

A translation job comprises one or multiple translation groups, as well as other information that resides at the job level and pertains to all translation items that are part of the translation job. A translation job is uniquely identified by the **JobId**. The information associated with a translation job is specified as follows:

- **JobId:** A bigint containing a unique identifier that identifies a translation job. This attribute MUST NOT be NULL.
- **Settings:** The xml containing settings to be applied across all translation items for the translation job. The meaning of the values in this field is defined by the protocol client. This attribute MUST NOT be NULL.
- **CreateTime:** A datetime that specifies when the translation job was created. This attribute MUST NOT be NULL.
- **CancelTime:** A datetime that specifies when the translation job was canceled.
- **Submitted:** A bit that specifies if the translation job has been submitted for translation. Translation items that belong to translation jobs that have not been submitted will not be translated. This attribute MUST NOT be NULL.
- **PartitionId:** A unique identifier that specifies the partition on which the translation job runs. The meaning of the values in this field is defined by the protocol client.
- **UserId:** An nvarchar(max) that contains the identifier of the user who submitted the translation job. The meaning of the values in this field is defined by the protocol client.
- **UserToken:** A varbinary(max) that contains the credentials of the user who submitted the translation job. The meaning of values in this field is defined by the protocol client. This attribute MUST NOT be NULL.
- **Name:** An nvarchar(max) that contains a label associated with the translation job that is specified by the protocol client. The meaning of the values in this field is defined by the protocol client.

### 3.1.2 Timers

None.

### 3.1.3 Initialization

None.

### 3.1.4 Higher-Layer Triggered Events

None.

### 3.1.5 Message Processing Events and Sequencing Rules

This interface includes the following methods:

Method	Description
proc_AddGroup, section <a href="#">3.1.5.1</a>	Adds a new translation group and the translation items that compose that translation group to the database.
proc_AddJob, section <a href="#">3.1.5.2</a>	Adds a new translation job to the database.
proc_CancelAllActiveJobs, section <a href="#">3.1.5.3</a>	Cancels all translation jobs that are active or unsubmitted.
proc_CancelJob, section <a href="#">3.1.5.4</a>	Cancels a translation job.
proc_GetConversionBatch, section <a href="#">3.1.5.5</a>	Retrieves a set of translation items that are not started or are already in progress but are stale.
proc_GetConversionBatchInterleaved, section <a href="#">3.1.5.6</a>	Retrieves a set of translation items that are not started or are already in progress but are stale.
proc_GetGroups, section <a href="#">3.1.5.7</a>	Gets a result set of translation groups of a translation job.
proc_GetItemCount, section <a href="#">3.1.5.8</a>	Gets the number of translation items added to the database in the past 24 hours.
proc_GetItems, section <a href="#">3.1.5.9</a>	Gets a result set of translation items in a translation job.
proc_GetJobs, section <a href="#">3.1.5.10</a>	Gets a list of translation jobs.
proc_GetJobStatus, section <a href="#">3.1.5.11</a>	Returns a result set specifying the status of the translation job.
proc_HasActiveJobs, section <a href="#">3.1.5.12</a>	Returns whether or not there are any active translation jobs.
proc_JobsExpire, section <a href="#">3.1.5.13</a>	Delete translation jobs.
proc_SubmitJob, section <a href="#">3.1.5.14</a>	Marks a translation job as entirely submitted so that the individual translation items can begin translation.
proc_UpdateConversionBatch, section <a href="#">3.1.5.15</a>	Update multiple translation items.
proc_UpdateFailedItem, section <a href="#">3.1.5.16</a>	Updates a translation item that has failed.
proc_UpdateItemCount, section <a href="#">3.1.5.17</a>	Add to the number of translation items added to the database in the past 24 hours.
proc_UpdateSucceededItem, section <a href="#">3.1.5.18</a>	Updates a translation item that has successfully finished.

The **T-SQL** syntax for each stored procedure and result set, and the variables of which they are composed are defined in [\[MS-TDS\]](#).

A protocol server SHOULD [<1>](#) verify that the inputs conform to the syntax specified in the following subsections. If the inputs do not conform to the syntax specified in the following subsections, then fail the call. Failure to verify inputs can introduce unpredictable inconsistencies.

### 3.1.5.1 proc\_AddGroup

The **proc\_AddGroup** stored procedure is called to add a new translation group and the translation items that compose that translation group to the database.

```
PROCEDURE proc_AddGroup (
    @JobId bigint
    ,@GroupId smallint
    ,@InputRoot nvarchar(max)
    ,@OutputRoot nvarchar(max)
    ,@MaxAttemptsCount smallint
    ,@JobXml xml
);
```

**@JobId:** The **JobId** identifying to which translation job the new translation group belongs. This parameter MUST NOT be NULL.

**@GroupId:** The **GroupId** of the new translation group. This parameter MUST NOT be NULL.

**@InputRoot:** The **InputRoot** of the new translation group. This parameter MUST NOT be NULL.

**@OutputRoot:** The **OutputRoot** of the new translation group. This parameter MUST NOT be NULL.

**@MaxAttemptsCount:** The initial **AttemptsRemaining** value for each of the new translation items. This parameter MUST NOT be NULL.

**@JobXml:** The xml that specifies the new translation items within the translation group to be added. This parameter MUST conform to the **databaseJobAdd** XML schema (section [2.2.6.4.2](#)). This parameter MUST NOT be NULL.

**Return Values:** The return value of this stored procedure MUST be ignored.

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

### 3.1.5.2 proc\_AddJob

The **proc\_AddJob** stored procedure is called to add a new translation job to the database. If the **@ItemCount** parameter is not NULL, the procedure checks that the translation item limit per 24-hour period, **@MaxItemsPerDay**, and the translation item limit per 24-hour period per **PartitionId**, **@MaxItemsPerPartitionPerDay**, will not be exceeded by the new translation job.

```
PROCEDURE proc_AddJob (
    @JobId bigint
    ,@UserId nvarchar(max) = null
    ,@UserToken varbinary(max)
    ,@PartitionId uniqueidentifier = null
    ,@Settings nvarchar(max)
    ,@Name nvarchar(max) = null
    ,@ItemCount int = null
    ,@MaxItemsPerDay int = null
    ,@MaxItemsPerPartitionPerDay int = null
);
```

**@JobId:** The **JobId** of the new translation job. This parameter MUST NOT be NULL.

**@UserId:** The **UserId** of the new translation job.

**@UserToken:** The **UserToken** of the new translation job. This parameter MUST NOT be NULL.

**@PartitionId:** The **PartitionId** of the new translation job.

**@Settings:** The **Settings** of the new translation job. This parameter MUST NOT be NULL.

**@Name:** The **Name** of the new translation job.

**@ItemCount:** The number of translation items which the job will contain.

**@MaxItemsPerDay:** The maximum number of translation items which can be added in a 24-hour period. A NULL value indicates that there is no limit.

**@MaxItemsPerPartitionPerDay:** The maximum number of translation items which can be added in a 24-hour period for each unique **PartitionId**. A NULL value indicates that there is no limit.

**Return Values:** An integer which MUST be in the following table.

Value	Description
1	Indicates that the translation job was not added because it would cause either <b>@MaxItemsPerDay</b> or <b>@MaxItemsPerPartitionPerDay</b> to be exceeded.
0	Indicates that the translation job was added.

**Result Sets:**

This procedure MUST NOT return any result sets.

### 3.1.5.3 proc\_CancelAllActiveJobs

The **proc\_CancelAllActiveJobs** stored procedure is called to cancel all translation jobs that are active or unsubmitted. This stored procedure updates the **CancelTime** to be the current time for any translation job that is either active or unsubmitted.

An active translation job has the following properties:

- The **CancelTime** of the translation job is NULL.
- The translation job contains translation items that either have a NULL **StartTime** or a NULL **StopTime**.

An unsubmitted translation job is a translation job that has the following properties:

- The **CancelTime** of the translation job is NULL.
- The **Submitted** of the translation job is zero.

```
PROCEDURE proc_CancelAllActiveJobs (  
);
```

**Return Values:** The return value of this stored procedure MUST be ignored.

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

### 3.1.5.4 proc\_CancelJob

The **proc\_CancelJob** stored procedure is called to cancel a translation job. This stored procedure changes the **CancelTime** of the translation job to be the current time.

The following conditions MUST be true for the translation job to be canceled.

- The **JobId** of the translation job is equal to *@JobId*.
- The **PartitionId** of the translation job is equal to *@PartitionId*.
- The **CancelTime** of the translation job is NULL.

```
PROCEDURE proc_CancelJob (  
    @JobId bigint  
    ,@PartitionId uniqueidentifier = null  
);
```

**@JobId:** The **JobId** of the translation job to be canceled. This parameter MUST NOT be NULL.

**@PartitionId:** The **PartitionId** of the translation job to be canceled.

**Return Values:** The return value of this stored procedure MUST be ignored.

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

### 3.1.5.5 proc\_GetConversionBatch

The **proc\_GetConversionBatch** stored procedure is called to retrieve a set of translation items that are not started or are already in progress but are stale.

A translation item that is not started have the following properties:

- the translation item's translation job has a **Submitted** equal to one.
- the translation item's translation job has a **CancelTime** that is NULL.
- the translation item has a **StartTime** that is NULL.
- the translation item has a **StopTime** that is NULL.

A translation item that is already in progress but is stale has the following properties:

- The translation item's translation job has a **Submitted** equal to one.
- The translation item's translation job has a **CancelTime** that is NULL.
- The translation item has a **StopTime** that is NULL.
- The translation item has a **StartTime** that is not NULL.
- The translation item has a **StartTime** that is less than **@InProgressThreshold**.

This stored procedure MUST return the number of rows representing not started translation items equal to **@NumberOfConversionsInBatch** in the result set if there are enough translation items in the database that meet the criteria. If there are not enough, every translation item that meets the criteria is returned in the result set.

```
PROCEDURE proc_GetConversionBatch (
    @NumberOfConversionsInBatch int
    ,@InProgressThreshold datetime
);
```

**@NumberOfConversionsInBatch:** The maximum number of not started translation items that can be returned by this stored procedure. This parameter MUST NOT be NULL.

**@InProgressThreshold:** The time threshold that specifies which translation items are considered stale. This parameter MUST NOT be NULL.

**Return Values:** The return value of this stored procedure MUST be ignored.

**Result Sets:**

This stored procedure MUST return a [proc\\_GetConversionBatch.ResultSet0](#)

### 3.1.5.6 proc\_GetConversionBatchInterleaved

The **proc\_GetConversionBatchInterleaved** stored procedure is called to retrieve a set of translation items that are not started or are already in progress but are stale.

A translation item that is not started has the following properties:

- the translation item's translation job has a **Submitted** equal to one.
- the translation item's translation job has a **CancelTime** that is NULL.
- the translation item has a **StartTime** that is NULL.
- the translation item has a **StopTime** that is NULL.

A translation item that is already in progress and is stale has the following properties:

- the translation item's translation job has a **Submitted** equal to one.
- the translation item's translation job has a **CancelTime** that is NULL.
- the translation item has a **StopTime** that is NULL.
- the translation item has a **StartTime** that is not NULL.
- the translation item has a **StartTime** that is less than **@InProgressThreshold**.

This stored procedure MUST return the number of rows representing not started translation items equal to **@NumberOfConversionsInBatch** in the result set if there are enough translation items in the database that meet the criteria. If there are not enough, every translation item that meets the criteria is returned in the result set.

This procedure differs from **proc\_GetConversionBatch** by attempting to distribute the number of not started translation items which are fetched evenly among each **PartitionId** for which there are eligible translation items.

```
PROCEDURE proc_GetConversionBatchInterleaved (
    @NumberOfConversionsInBatch int
    ,@InProgressThreshold datetime
);
```

);

**@NumberOfConversionsInBatch:** The maximum number of not started translation items that can be returned by this stored procedure. This parameter MUST NOT be NULL.

**@InProgressThreshold:** The time threshold that specifies which translation items are considered stale. This parameter MUST NOT be NULL.

**Return Values:** The return value of this stored procedure MUST be ignored.

**Result Sets:**

This stored procedure MUST return a [proc\\_GetConversionBatchInterleaved.ResultSet0](#)

### 3.1.5.7 proc\_GetGroups

The **proc\_GetGroups** stored procedure is called to get a result set of translation groups of a translation job.

```
PROCEDURE proc_GetGroups (  
    @JobId bigint  
    ,@PartitionId uniqueidentifier = null  
);
```

**@JobId:** The **JobId** of the translation job. Any translation group returned by this stored procedure MUST have a **JobId** equal to **@JobId**. This parameter MUST NOT be NULL.

**@PartitionId:** The **PartitionId** of the translation job. If this is not NULL, any translation group returned by this stored procedure MUST belong to a translation job that has a **PartitionId** equal to **@PartitionId**.

**Return Values:** The return value of this stored procedure MUST be ignored.

**Result Sets:**

This stored procedure MUST return a [proc\\_GetGroups.ResultSet0](#)

### 3.1.5.8 proc\_GetItemCount

The **proc\_GetItemCount** stored procedure returns the number of translation items which have been added to the database in the past 24 hours for a given **PartitionId**, or in total.

```
PROCEDURE proc_GetItemCount (  
    @PartitionId uniqueidentifier  
    ,@Hour int  
    ,@Date date  
);
```

**@PartitionId:** The **PartitionId** of translation jobs for which translation items are counted. If this parameter is 00000000-0000-0000-0000-000000000000, all translation items are counted regardless of **PartitionId**. This parameter MUST NOT be NULL.

**@Hour:** An integer representing the current hour of day. This parameter MUST NOT be NULL and MUST be in the range 0 to 23, inclusive.



**@Date:** The current date. This parameter MUST NOT be NULL.

**Return Values:** An integer indicating a number of translation items.

**Result Sets:**

This procedure MUST NOT return any result sets.

### 3.1.5.9 proc\_GetItems

The **proc\_GetItems** stored procedure is called to get a result set of translation items in a translation job.

```
PROCEDURE proc_GetItems (  
    @JobId bigint  
    ,@GroupId smallint  
    ,@PartitionId uniqueidentifier = null  
    ,@NotSubmitted bit  
    ,@NotStarted bit  
    ,@InProgress bit  
    ,@Succeeded bit  
    ,@Failed bit  
    ,@Canceled bit  
);
```

**@JobId:** The **JobId** of the translation job. Any translation item returned by this stored procedure MUST have a **JobId** that matches **@JobId**. This parameter MUST NOT be NULL.

**@GroupId:** The **GroupId** of the translation item. Any translation item returned by this stored procedure MUST have a **GroupId** that matches **@GroupId**. This parameter MUST NOT be NULL.

**@PartitionId:** The **PartitionId** of the translation job. Any translation item returned by this stored procedure MUST belong to a translation job with a **PartitionId** that matches **@PartitionId**.

**@NotSubmitted:** If this parameter is equal to zero, the result set returned by this stored procedure MUST NOT contain any translation items that have the following:

- the translation item's translation job has a **Submitted** equal to zero.
- the translation item's translation job has a **CancelTime** that is NULL.

**@NotStarted:** If this parameter is equal to zero, the result set returned by this stored procedure MUST NOT contain any translation items that have the following:

- the translation item's translation job has a **Submitted** equal to one.
- the translation item's translation job has a **CancelTime** that is NULL.
- the translation item has a **StartTime** that is NULL.

**@InProgress:** If this parameter is equal to zero, the result set returned by this stored procedure MUST NOT contain any translation items that have the following:

- the translation item's translation job has a **Submitted** equal to one.
- the translation item's translation job has a **CancelTime** that is NULL.

- the translation item has a **StartTime** that is not NULL.
- the translation item has a **StopTime** that is NULL.

**@Succeeded:** If this parameter is equal to zero, the result set returned by this stored procedure MUST NOT contain any translation items that have the following:

- the translation item's translation job has a **Submitted** equal to one.
- the translation item has a **StartTime** that is not NULL.
- the translation item has a **StopTime** that is not NULL.
- the translation item has an **ErrorCode** that is NULL.

**@Failed:** If this parameter is equal to zero, the result set returned by this stored procedure MUST NOT contain any translation items that have the following:

- the translation item's translation job has a **Submitted** equal to one.
- the translation item has a **StartTime** that is not NULL.
- the translation item has a **StopTime** that is not NULL.
- the translation item has an **ErrorCode** that is not NULL.

**@Canceled:** If this parameter is equal to zero, the result set returned by this stored procedure MUST NOT contain any translation items that have the following:

- the translation item's translation job has a **CancelTime** that is not NULL.
- the translation item has a **StopTime** that is NULL.

**Return Values:** The return value of this stored procedure MUST be ignored.

**Result Sets:**

This stored procedure MUST return a [proc\\_GetItems.ResultSet0](#)

### 3.1.5.10 proc\_GetJobs

The **proc\_GetJobs** stored procedure is called to get a list of translation jobs.

```
PROCEDURE proc_GetJobs (
  @PartitionId uniqueidentifier = null
  ,@UserId nvarchar(max) = null
  ,@ActiveOnly bit
  ,@SubmittedOnly bit
);
```

**@PartitionId:** Any translation job returned by this stored procedure MUST have a **PartitionId** that equals this parameter.

**@UserId:** If this parameter is not NULL, any translation job returned by this stored procedure MUST have a **UserId** that equals this parameter.

**@ActiveOnly:** If this parameter is equal to one, any translation job returned by this stored procedure MUST have the following:

- A **CancelTime** that is NULL.
- There is at least one translation item that belongs to the translation job with a **StopTime** that is NULL.

**@SubmittedOnly:** If this parameter is equal to one, any translation job returned by this stored procedure MUST have a **Submitted** that equals one.

**Return Values:** The return value of this stored procedure MUST be ignored.

**Result Sets:**

This stored procedure MUST return a [proc\\_GetJobs.ResultSet0](#)

### 3.1.5.11 proc\_GetJobStatus

The **proc\_GetJobStatus** stored procedure returns a result set specifying the status of the translation job.

```
PROCEDURE proc_GetJobStatus (  
  @JobId bigint  
  ,@PartitionId uniqueidentifier = null  
);
```

**@JobId:** The **JobId** of the translation job that this stored procedure will return.

**@PartitionId:** The **PartitionId** of the translation job that this stored procedure will return. The translation job identified by **JobId** MUST NOT be described in the result set unless it has a **PartitionId** that equals **@PartitionId**.

**Return Values:** The return value of this stored procedure MUST be ignored.

**Result Sets:**

This stored procedure MUST return a [proc\\_GetJobStatus.ResultSet0](#)

### 3.1.5.12 proc\_HasActiveJobs

The **proc\_HasActiveJobs** stored procedure indicates whether or not there are any active translation jobs. An active translation job is a translation job that has the following properties:

- **Submitted** is 1.
- **CancelTime** is NULL.
- There is at least one translation item with a NULL **StopTime**.

```
PROCEDURE proc_HasActiveJobs (  
);
```

**Return Values:** An integer which MUST be in the following table.

Value	Description
0	There are no active translation jobs.
Jobs.count	There is at least one active translation job.

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

### 3.1.5.13 proc\_JobsExpire

The **proc\_JobsExpire** stored procedure is called to delete translation jobs. Deleting a translation job also deletes all translation groups and translation items associated with the translation job.

This stored procedure deletes translation jobs according to the following criteria:

- In all cases, at least one of the following conditions MUST be true for a translation job, or any translation items pertaining to it, to be deleted:
  - the translation job's **PartitionId** is equal to **@PartitionId**.
  - @AllPartitions** equals one.
- If **@JobId** is not NULL, the translation job with a **JobId** matching **@JobId** is deleted.
- Otherwise, if **@IncludeActiveJobs** is one, all translation jobs meeting the **CreateTime criterion** are deleted.
- Otherwise,
  - All translation jobs meeting the **CancelTime criterion** are deleted.
  - If **@TimeThreshold** is not NULL, all translation jobs meeting the **CreateTime criterion** and **Submitted** is equal to zero are deleted.
  - All translation items meeting the **StopTime criterion** are deleted.
  - All translation jobs for which all translation items meet the **StopTime criterion** are deleted.
- The **CreateTime criterion** is that either **@TimeThreshold** is NULL or the translation job's **CreateTime** is less than **@TimeThreshold**.
- The **CancelTime criterion** is that the translation job's **CancelTime** is not NULL, and that either **@TimeThreshold** is NULL or the translation job's **CancelTime** is less than **@TimeThreshold**.
- The **StopTime criterion** is that the translation item's **StopTime** is not NULL, and that either **@TimeThreshold** is NULL or the translation item's **StopTime** is less than **@TimeThreshold**.

```

PROCEDURE proc_JobsExpire (
  @TimeThreshold datetime = null
  ,@PartitionId uniqueidentifier = null
  ,@AllPartitions bit = null
  ,@JobId bigint = null
  ,@IncludeActiveJobs bit
);

```

**@TimeThreshold:** Translation jobs created and completed before this time are eligible for deletion.

**@PartitionId:** Translation jobs with an equivalent **PartitionId** are eligible for deletion.

**@AllPartitions:** This parameter specifies whether or not translation jobs are eligible for deletion regardless of **PartitionId**.

**@JobId:** The translation job with an equivalent **JobId** is eligible for deletion.

**@IncludeActiveJobs:** This parameter specifies whether or not active translation jobs are eligible for deletion. This parameter MUST NOT be NULL.

**Return Values:** The return value of this stored procedure MUST be ignored.

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

#### 3.1.5.14 **proc\_SubmitJob**

The **proc\_SubmitJob** stored procedure is called to mark a translation job as entirely submitted so that the individual translation items can begin translation. Any translation job identified by **@JobId** MUST have **Submitted** set to one as a result of calling this stored procedure.

```
PROCEDURE proc_SubmitJob (  
    @JobId bigint  
);
```

**@JobId:** The **JobId** of the translation job to update. This parameter MUST NOT be NULL.

**Return Values:** The return value of this stored procedure MUST be ignored.

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

#### 3.1.5.15 **proc\_UpdateConversionBatch**

The **proc\_UpdateConversionBatch** stored procedure is called to update multiple translation items. This stored procedure also returns a result set that contains information about the translation jobs and translation groups that contain translation items that were updated by this stored procedure.

The **@BatchXml** MUST conform to the **databaseBatchUpdate** XML schema, as specified in section [2.2.6.4.1](#). The **@BatchXml** contains two elements, **batch.failed** and **batch.start**.

For every element under **batch.start** in **@BatchXml** that meets the following criteria:

- the **JobId** of the translation item equals **item.job**.
- the **GroupId** of the translation item equals **item.group**.
- the **ItemId** of the translation item equals **item.id**.

This stored procedure updates those translation items as follows:

- the **StartTime** of the translation item is set to the current time.
- the **WorkerServerInstance** of the translation item is set to **item.wsi**.
- the **AttemptsRemaining** of the translation item is decremented by one, but cannot be less than 0.

For every element under **batch.failed** in **@BatchXml** that meets the following criteria:

- the **JobId** of the translation item equals **item.job**.
- the **GroupId** of the translation item equals **item.group**.
- the **ItemId** of the translation item equals **item.id**.

This stored procedure updates those translation items as follows:

- the **WorkerServerInstance** of the translation item is set to NULL.
- the **ErrorCode** of the translation item is set to **item.error**.
- the **StopTime** of the translation item is set to the current time.

```
PROCEDURE proc_UpdateConversionBatch (
  @BatchXml xml
);
```

**@BatchXml:** The translation items to be updated and how they are updated. This parameter MUST NOT be NULL. This parameter MUST conform to the **databaseBatchUpdate** XML schema, as specified in section [2.2.6.4.1](#).

**Return Values:** The return value of this stored procedure MUST be ignored.

**Result Sets:**

This stored procedure MUST return a [proc\\_UpdateConversionBatch.ResultSet0](#)

### 3.1.5.16 proc\_UpdateFailedItem

The **proc\_UpdateFailedItem** stored procedure is called to update a translation item that has failed.

If **@NoRetry** is zero or **AttemptsRemaining** of the translation item is greater than zero, the translation item identified by **@JobId**, **@GroupId**, and **@ItemId** is updated as follows:

- the **StartTime** is set to NULL.
- the **WorkerServerInstance** is set to NULL.

Otherwise, the translation item is updated as follows:

- the **StopTime** is set to the current time.
- the **ErrorCode** is set to **@ErrorCode**.
- the **WorkerServerInstance** is set to NULL.
- the **AttemptsRemaining** is set to zero.

```
PROCEDURE proc_UpdateFailedItem (
  @JobId bigint
  ,@GroupId smallint
  ,@ItemId int
  ,@NoRetry bit
  ,@ErrorCode int
```

);

**@JobId:** The **JobId** of the translation item to update. This parameter MUST NOT be NULL.

**@GroupId:** The **GroupId** of the translation item to update. This parameter MUST NOT be NULL.

**@ItemId:** The **ItemId** of the translation item to update. This parameter MUST NOT be NULL.

**@NoRetry:** Indicates if translation of the translation item can be attempted again.

**@ErrorCode:** The new value for the **ErrorCode** field of the translation item.

**Return Values:** The return value of this stored procedure MUST be ignored.

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

### 3.1.5.17 **proc\_UpdateItemCount**

The **proc\_UpdateItemCount** stored procedure is called to add to the count of translation items which have been added to the database in the past 24 hours. The count is incremented in **@ItemCount**.

```
PROCEDURE proc_UpdateItemCount (  
    @PartitionId uniqueidentifier  
    ,@Hour int  
    ,@Date date  
    ,@ItemCount int  
);
```

**@PartitionId:** The **PartitionId** of the translation job to which translation items are to be added. If the **PartitionId** of the translation job is NULL, this parameter MUST be 00000000-0000-0000-0000-000000000000. This parameter MUST NOT be NULL.

**@Hour:** An integer representing the current hour of day. This parameter MUST NOT be NULL, and MUST be in the range 0 to 23, inclusive.

**@Date:** The current date. This parameter MUST NOT be NULL.

**@ItemCount:** The number of translation items which will be added. This parameter MUST NOT be NULL.

**Return Values:** The return value of this stored procedure MUST be ignored.

**Result Sets:**

This procedure MUST NOT return any result sets.

### 3.1.5.18 **proc\_UpdateSucceededItem**

The **proc\_UpdateSucceededItem** stored procedure is called to update a translation item that has successfully finished. This stored procedure MUST do the following:

- the **StopTime** of the translation item is set to the current time.
- the **ErrorCode** of the translation item is set to NULL.

- the **WorkerServerInstance** of the translation item is set to NULL.

```
PROCEDURE proc_UpdateSucceededItem (  
    @JobId bigint  
    ,@GroupId smallint  
    ,@ItemId int  
);
```

**@JobId:** The **JobId** of the translation item to update. This parameter MUST NOT be NULL.

**@GroupId:** The **GroupId** of the translation item to update. This parameter MUST NOT be NULL.

**@ItemId:** The **ItemId** of the translation item to update. This parameter MUST NOT be NULL.

**Return Values:** The return value of this stored procedure MUST be ignored.

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

### 3.1.6 Timer Events

None.

### 3.1.7 Other Local Events

None.

## 3.2 Client Details

None.



## 4 Protocol Examples

This example describes the requests that are made when the protocol client adds a translation job to the database and translates the translation items. The example assumes the database begins with no existing translation jobs or translation items.

In this example, the steps occur in the following order:

```
(1)-- proc_AddJob -->
(2)Protocol client confirms that translation job was added.
(3)-- proc_AddGroup -->
(4)<-- return code ignored --
(5)-- proc_SubmitJob -->
(6)<-- return code ignored --
(7)-- proc_GetConversionBatch -->
(8) <-- proc_GetConversionBatch Result Set is returned --
(9)-- proc_UpdateConversionBatch -->
(10) <-- proc_UpdateConversionBatch Result Set is returned --
(11)Protocol client translates the translation items.
(12)-- proc_UpdateSucceededItem -->
(13)<-- return code ignored --
(14)-- proc_UpdateFailedItem -->
(15)<-- return code ignored --
```

**Step 1:** The client creates a translation job and adds it to the database on the protocol server. This is done by calling the **proc\_AddJob** stored procedure (section [3.1.5.2](#)) using [\[MS-TDS\]](#).

Consider the following T-SQL syntax, which displays the parameters used to call this stored procedure:

```
exec dbo.proc_AddJob
@JobId = '1',
@UserId = 'user@contoso.com',
@UserToken = 0x1234567890,
@PartitionId = '93572c0a-d9e1-1395-dab3-932eac7ba30c',
@Settings = '<settings/>',
@Name = 'testJob',
@ItemCount = '2',
@MaxItemsPerDay = '1000',
@MaxItemsPerPartitionPerDay = '100'
```

**Step 2:** The protocol server responds with a return code of zero, which indicates to the client that the translation job was added.

**Step 3:** The client then adds translation items to the translation job by calling the **proc\_AddGroup** stored procedure (section [3.1.5.1](#)) using [\[MS-TDS\]](#).

Consider the following T-SQL syntax, which displays the parameters used to call this stored procedure:

```
exec dbo.proc_AddGroup
@JobId = '1',
@GroupId = '1',
@InputRoot = 'http://contoso.com',
@OutputRoot = 'http://contoso.com',
```

```

@MaxAttemptsCount = '2',
@JobXml =
'<group
xmlns="http://schemas.microsoft.com/office/server/translation/2010/11/databaseJobAdd">
  <item id="1" in="Aenean%20nec.docx" out="Aenean%20nec_Thai.docx" />
  <item id="2" in="Fusce%20aliquet.docx" out="Fusce%20aliquet_Thai.docx" />
</group>'

```

**Step 4:** The protocol server returns a return code, which is ignored by the client.

**Step 5:** When the client has no more translation items to add, the client calls the **proc\_SubmitJob** stored procedure (section [3.1.5.14](#)) using [MS-TDS].

Consider the following T-SQL syntax, which displays the parameters used to call this stored procedure:

```

exec dbo.proc_SubmitJob
@JobId = '1'

```

**Step 6:** The protocol server returns a return code, which is ignored by the client.

**Step 7:** The client queries the protocol server for which translation items are to be run next. The client will query for two items from the protocol server. It does this by calling the **proc\_GetConversionBatch** stored procedure (section [3.1.5.5](#)) by using the [MS-TDS].

Consider the following T-SQL syntax, which displays the parameters used to call this stored procedure:

```

exec dbo.proc_GetConversionBatch
@NumberOfConversionsInBatch = 2,
@InProgressThreshold = 'Jan 31 2008 01:01:01:000AM'

```

**Step 8:** The protocol server returns the **proc\_GetConversionBatch.ResultSet0** (section [2.2.4.1](#)). The result set contains the following fields:

Field Name	Row 1	Row 2
JobId	1	1
GroupId	1	1
ItemId	1	2
InProgress	0	0
InputFile	Aenean%20nec.docx	Fusce%20aliquet.docx
OutputFile	Aenean%20nec_Thai.docx	Fusce%20aliquet_Thai.docx
AttemptsRemaining	2	2
WorkerServerInstance	NULL	NULL
StartTime	NULL	NULL

Field Name	Row 1	Row 2
CreateTime	2010-01-19 14:18:53.397	2010-01-19 14:18:53.397

**Step 9:** The protocol client updates the translation items with the unique identifier of the **WorkerServerInstance**, and marks the translation items to be started. The client does this by calling the **proc\_UpdateConversionBatch** stored procedure (section 3.1.5.15) using [MS-TDS].

Consider the following T-SQL syntax, which displays the parameters used to call this stored procedure:

```
exec dbo.proc_UpdateConversionBatch
@BatchXml = '<batch
xmlns="http://schemas.microsoft.com/office/server/translation/2010/11/databaseBatchUpdate">
  <start>
    <item job="1" group="1" id="1" wsi="b00ae9a1-0474-474e-b348-f6a8bcc95331" />
    <item job="1" group="1" id="2" wsi="b00ae9a1-0474-474e-b348-f6a8bcc95331" />
  </start>
  <failed />
</batch>'
```

**Step 10:** The protocol server returns the **proc\_UpdateConversionBatch.ResultSet0** (section 2.2.4.7) which contains only one row in this case because there was only one translation job that all the items belonged to:

Field Name	Value
JobId	1
GroupId	1
InputRoot	http://contoso.com
OutputRoot	http://contoso.com
Settings	<settings/>
PartitionId	93572c0a-d9e1-1395-dab3-932eac7ba30c
UserToken	0x1234567890

**Step 11:** The protocol client begins file translation with the information from steps 8 and 10. With the **proc\_GetConversionBatch.ResultSet0** from step 8, the client has the identity of the translation item through the **JobId**, **GroupId**, and **ItemId** parameters, as well as the source and destination file locations with the **InputFile** and **OutputFile** parameters. With the **proc\_UpdateConversionBatch.ResultSet0** from step 10, the client has additional information: the user identity (**UserId**) and credentials (**UserToken**), any additional translation settings (**Settings**) for the translation job, the **JobId** and **GroupId** of the translation items, and the **InputRoot** and **OutputRoot**.

The translation item identified uniquely by (**JobId = 1, GroupId = 1, ItemId = 1**) finishes translating first and sometime later the translation item (**JobId = 1, GroupId = 1, ItemId = 2**) finishes but fails to translate because the document is corrupt.

**Step 12:** Upon completion of the translation item (**JobId** = 1, **GroupId** = 1, **ItemId** = 1), the protocol client calls the **proc\_UpdateSucceededItem** stored procedure (section [3.1.5.18](#)) because the translation item succeeded in translation.

Consider the following T-SQL syntax which displays the parameters used to call this stored procedure:

```
exec dbo.proc_UpdateSucceededItem
@JobId = '1',
@GroupId = '1',
@ItemId = '1'
```

**Step 13:** The protocol server returns a return code, which is ignored by the client.

**Step 14:** Upon completion of the translation item (**JobId** = 1, **GroupId** = 1, **ItemId** = 2), the protocol client calls the **proc\_UpdateFailedItem** stored procedure (section [3.1.5.16](#)) because the translation item failed due the file being corrupt. **@NoRetry** is set to one because the protocol client has determined that the document is corrupt and will not be for an attempted translation again.

Consider the following T-SQL syntax which displays the parameters used to call this stored procedure:

```
exec dbo.proc_UpdateFailedItem
@JobId = '1',
@GroupId = '1',
@ItemId = '2',
@ErrorCode = '1',
@NoRetry = '1'
```

**Step 15:** The protocol server returns a return code, which is ignored by the client.

## 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 prior to invoking the stored procedure.

There are no additional security considerations for implementers. Security assumptions are documented in section [1.5](#).

### 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® 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.

[<1> Section 3.1.5:](#) SharePoint Server 2013 Preview does not verify the inputs.

## 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  
    [server](#) 17  
[Applicability](#) 7  
[Attribute groups - overview](#) 16  
[Attributes - overview](#) 16

### B

[Binary structures - overview](#) 8  
[Bit fields - overview](#) 8

### C

[Canonical example](#) 33  
[Capability negotiation](#) 7  
[Change tracking](#) 39  
[Common data types](#) 8  
[Complex types - overview](#) 13

### D

Data model - abstract  
    [server](#) 17  
Data types - simple  
    [overview](#) 8  
databaseBatchUpdate  
    [element](#) 13  
databaseJobAdd  
    [element](#) 15

### E

Elements  
    [databaseBatchUpdate](#) 13  
    [databaseJobAdd](#) 15  
[Elements - overview](#) 13  
Events  
    [local - server](#) 32  
    [timer - server](#) 32  
Examples  
    [canonical](#) 33

### F

[Fields - vendor-extensible](#) 7  
[Flag structures - overview](#) 8

### G

[Glossary](#) 5  
[Groups - overview](#) 16

### H

Higher-layer triggered events  
    [server](#) 18

### I

[Implementer - security considerations](#) 37  
[Index of security parameters](#) 37  
[Informative references](#) 6  
Initialization  
    [server](#) 18  
[Introduction](#) 5

### L

Local events  
    [server](#) 32

### M

Message processing  
    [server](#) 19  
Messages  
    [attribute groups](#) 16  
    [attributes](#) 16  
    [binary structures](#) 8  
    [bit fields](#) 8  
    [complex types](#) 13  
    [databaseBatchUpdate element](#) 13  
    [databaseJobAdd element](#) 15  
    [elements](#) 13  
    [enumerations](#) 8  
    [flag structures](#) 8  
    [groups](#) 16  
    [namespaces](#) 13  
    [result sets](#) 8  
    [simple data types](#) 8  
    [simple types](#) 13  
    [table and views](#) 12  
    [table structures](#) 12  
    [transport](#) 8  
    [view structures](#) 12  
    [XML structures](#) 13  
Methods  
    [proc\\_AddGroup](#) 20  
    [proc\\_AddJob](#) 20  
    [proc\\_CancelAllActiveJobs](#) 21  
    [proc\\_CancelJob](#) 22  
    [proc\\_GetConversionBatch](#) 22  
    [proc\\_GetConversionBatchInterleaved](#) 23  
    [proc\\_GetGroups](#) 24  
    [proc\\_GetItemCount](#) 24  
    [proc\\_GetItems](#) 25  
    [proc\\_GetJobs](#) 26  
    [proc\\_GetJobStatus](#) 27  
    [proc\\_HasActiveJobs](#) 27  
    [proc\\_JobsExpire](#) 28  
    [proc\\_SubmitJob](#) 29  
    [proc\\_UpdateConversionBatch](#) 29  
    [proc\\_UpdateFailedItem](#) 30  
    [proc\\_UpdateItemCount](#) 31  
    [proc\\_UpdateSucceededItem](#) 31



## N

[Namespaces](#) 13  
[Normative references](#) 5

## O

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

## P

[Parameters - security index](#) 37  
[Preconditions](#) 7  
[Prerequisites](#) 7  
[proc\\_AddGroup method](#) 20  
[proc\\_AddJob method](#) 20  
[proc\\_CancelAllActiveJobs method](#) 21  
[proc\\_CancelJob method](#) 22  
[proc\\_GetConversionBatch method](#) 22  
[proc\\_GetConversionBatch.ResultSet0 result set](#)  
([section 2.2.4.1](#) 8, [section 2.2.4.2](#) 9)  
[proc\\_GetConversionBatchInterleaved method](#) 23  
[proc\\_GetGroups method](#) 24  
[proc\\_GetGroups.ResultSet0 result set](#) 9  
[proc\\_GetItemCount method](#) 24  
[proc\\_GetItems method](#) 25  
[proc\\_GetItems.ResultSet0 result set](#) 10  
[proc\\_GetJobs method](#) 26  
[proc\\_GetJobs.ResultSet0 result set](#) 10  
[proc\\_GetJobStatus method](#) 27  
[proc\\_GetJobStatus.ResultSet0 result set](#) 11  
[proc\\_HasActiveJobs method](#) 27  
[proc\\_JobsExpire method](#) 28  
[proc\\_SubmitJob method](#) 29  
[proc\\_UpdateConversionBatch method](#) 29  
[proc\\_UpdateConversionBatch.ResultSet0 result set](#)  
12  
[proc\\_UpdateFailedItem method](#) 30  
[proc\\_UpdateItemCount method](#) 31  
[proc\\_UpdateSucceededItem method](#) 31  
[Product behavior](#) 38

## R

[References](#) 5  
[informative](#) 6  
[normative](#) 5  
[Relationship to other protocols](#) 6  
Result sets  
[proc\\_GetConversionBatch.ResultSet0](#) 8  
[proc\\_GetConversionBatchInterleaved.ResultSet0](#)  
9  
[proc\\_GetGroups.ResultSet0](#) 9  
[proc\\_GetItems.ResultSet0](#) 10  
[proc\\_GetJobs.ResultSet0](#) 10  
[proc\\_GetJobStatus.ResultSet0](#) 11  
[proc\\_UpdateConversionBatch.ResultSet0](#) 12  
[Result sets - overview](#) 8

## S

Security

[implementer considerations](#) 37  
[parameter index](#) 37

Sequencing rules  
[server](#) 19

## Server

[abstract data model](#) 17  
[higher-layer triggered events](#) 18  
[initialization](#) 18  
[local events](#) 32  
[message processing](#) 19  
[proc\\_AddGroup method](#) 20  
[proc\\_AddJob method](#) 20  
[proc\\_CancelAllActiveJobs method](#) 21  
[proc\\_CancelJob method](#) 22  
[proc\\_GetConversionBatch method](#) 22  
[proc\\_GetConversionBatchInterleaved method](#) 23  
[proc\\_GetGroups method](#) 24  
[proc\\_GetItemCount method](#) 24  
[proc\\_GetItems method](#) 25  
[proc\\_GetJobs method](#) 26  
[proc\\_GetJobStatus method](#) 27  
[proc\\_HasActiveJobs method](#) 27  
[proc\\_JobsExpire method](#) 28  
[proc\\_SubmitJob method](#) 29  
[proc\\_UpdateConversionBatch method](#) 29  
[proc\\_UpdateFailedItem method](#) 30  
[proc\\_UpdateItemCount method](#) 31  
[proc\\_UpdateSucceededItem method](#) 31  
[sequencing rules](#) 19  
[timer events](#) 32  
[timers](#) 18

## Simple data types

[overview](#) 8  
[Simple types - overview](#) 13  
[Standards assignments](#) 7

## Structures

[binary](#) 8  
[table and view](#) 12  
[XML](#) 13

## T

[Table structures - overview](#) 12  
Tables and views  
[overview](#) 12  
Timer events  
[server](#) 32  
Timers  
[server](#) 18  
[Tracking changes](#) 39  
[Transport](#) 8  
Triggered events - higher-layer  
[server](#) 18  
Types  
[complex](#) 13  
[simple](#) 13

## V

[Vendor-extensible fields](#) 7  
[Versioning](#) 7  
[View structures - overview](#) 12

**x**

[XML structures](#) 13

Preliminary