[MS-WORDSSP]:
Word Automation Services Stored Procedures Protocol Specification

Revision Summary

Date	Revision History	Revision Class	Comments
07/13/2009	0.1	Major	Initial Availability
08/28/2009	0.2	Editorial	Revised and edited the technical content
11/06/2009	0.3	Editorial	Revised and edited the technical content
02/19/2010	1.0	Major	Updated and revised the technical content
03/31/2010	1.01	Editorial	Revised and edited the technical content
04/30/2010	1.02	Editorial	Revised and edited the technical content
06/07/2010	1.03	Editorial	Revised and edited the technical content
06/29/2010	1.04	Editorial	Changed language and formatting in the technical content.
07/23/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
09/27/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
06/10/2011	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
01/20/2012	1.5	Minor	Clarified the meaning of the technical content.
04/11/2012	1.5	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	1.5	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction

This document specifies the Word Services Stored Procedures Protocol. This protocol allows clients to store on the server information about converting documents from one file format 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]:

Coordinated Universal Time (UTC)
Security Support Provider Interface (SSPI)

The following terms are defined in [MS-OFCGLOS]:

conversion group
conversion item
conversion job
result set
return code
SQL authentication
stored procedure
Structured Query Language (SQL)
Transact-Structured Query Language (T-SQL)
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. 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

[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".

[MS-WORDSWCF] Microsoft Corporation, "Word Automation Services WCF Service Protocol Specification".

1.3 Protocol Overview (Synopsis)

This protocol allows clients to add, modify, and delete **conversion job**s, **conversion group**s and **conversion item**s from a database on the protocol server, as well as retrieve those conversion jobs, conversion groups and conversion 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 conversion jobs, conversion groups, and conversion items. The stored procedures do this via **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:

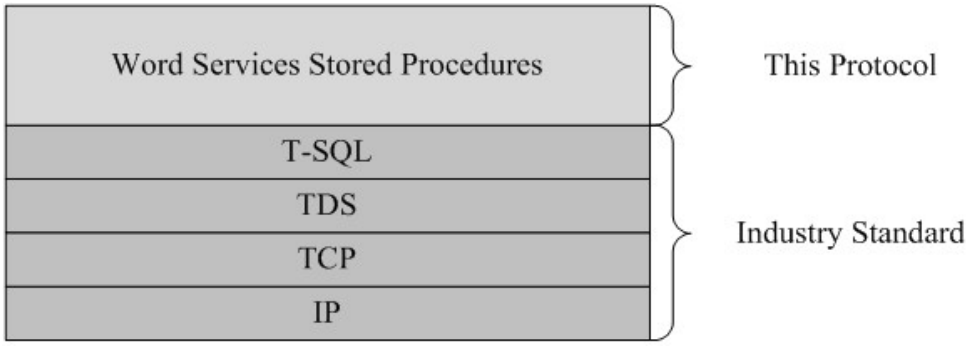

Figure 1: This protocol in relation to other protocols

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 has 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 conversion jobs, conversion groups, and conversion 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

[MS-TDS] 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_UpdateConversionBatch.ResultSet0

The **proc_UpdateConversionBatch.ResultSet0** result set contains information about the conversion jobs and conversion groups for the conversion items that were updated in the proc_UpdateConversionBatch stored procedure.

```
JobId bigint,
GroupId smallint,
InputRoot nvarchar(max),
OutputRoot nvarchar(max),
Settings nvarchar(max),
UserTokenHeader varbinary(32),
UserTokenSid varbinary(max),
UserTokenGroups varbinary(max),
```

JobId: The **JobId** of the conversion job.

GroupId: The **GroupId** of the conversion group.

InputRoot: The **InputRoot** of the conversion group.

OutputRoot: The **OutputRoot** of the conversion group.

Settings: The **Settings** of the conversion job.

UserTokenHeader: The **UserTokenHeader** of the conversion job.

UserTokenSid: The **UserTokenSid** of the conversion job.

UserTokenGroups: The **UserTokenGroups** of the conversion job.

2.2.4.2 proc_GetJobStatus.ResultSet0

The **proc_GetJobStatus.ResultSet0** result set contains information about the conversion items in a conversion job.

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

Total: This is the total number of conversion items in the conversion job.

NotSubmitted: This is the number of conversion items in the conversion job that have the following:

- The conversion job's **Submitted** is 0.
- The conversion job's **CancelTime** is NULL.

NotStarted: This is the number of conversion items in the conversion job that have the following:

- The conversion job's **Submitted** is 1.
- The conversion job's **CancelTime** is NULL.
- The conversion item's **StartTime** is NULL.

InProgress: This is the number of conversion items in the conversion job that have the following:

- The conversion job's **Submitted** is 1.
- The conversion job's **CancelTime** is NULL.
- The conversion item's **StartTime** is not NULL.
- The conversion item's **StopTime** is NULL.

Succeeded: This is the number of conversion items in the conversion job that have the following:

- The conversion job's **Submitted** is 1.
- The conversion item's **StartTime** is not NULL.
- The conversion item's **StopTime** is not NULL.
- The conversion item's **ErrorCode** is NULL.

Failed: This is the number of conversion items in the conversion job that have the following:

- The conversion job's **Submitted** is 1.
- The conversion item's **StartTime** is not NULL.
- The conversion item's **StopTime** is not NULL.

- The conversion item's **ErrorCode** is not NULL

Canceled: This is the number of conversion items in the conversion job that have the following:

- The conversion job's **Submitted** is 1.
- The conversion job's **CancelTime** is not NULL.
- The conversion item's **StopTime** is NULL.

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

2.2.4.3 proc_GetJobs.ResultSet0

The **proc_GetJobs.ResultSet0** result set contains information about a conversion job. Each row represents a conversion job and information about it.

```
JobId bigint,
CreateTime datetime,
CancelTime datetime,
Submitted bit,
Name nvarchar(max),
```

JobId: The **JobId** of a conversion job.

CreateTime: The **CreateTime** of a conversion job.

CancelTime: The **CancelTime** of a conversion job.

Submitted: The **Submitted** of a conversion job.

Name: The **Name** of a conversion job.

2.2.4.4 proc_GetItems.ResultSet0

The **proc_GetJobs.ResultSet0** result set contains information about conversion items in a conversion job. Each row represents a conversion item.

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

ItemId: The conversion item's **ItemId**.

StartTime: The conversion item's **StartTime**.

StopTime: The conversion item's **StopTime**.

ErrorCode: The conversion item's **ErrorCode**.

InputFile: The conversion item's **InputFile**.

OutputFile: The conversion item's **OutputFile**.

2.2.4.5 proc_GetGroups.ResultSet0

The **proc_GetGroups.ResultSet0** result set contains information about conversion groups in a conversion job. Each row represents a conversion group and also contains additional information from the conversion job.

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

GroupId: The conversion group's **GroupId**.

InputRoot: The conversion group's **InputRoot**.

OutputRoot: The conversion group's **OutputRoot**.

CreateTime: The **CreateTime** of the conversion job specified by the **@JobId** parameter.

CancelTime: The **CancelTime** of the conversion job specified by the **@JobId** parameter.

Submitted: The **Submitted** of the conversion job specified by the **@JobId** parameter.

Settings: The **Settings** of the conversion job specified by the **@JobId** parameter.

2.2.4.6 proc_GetConversionBatch.ResultSet0

The **proc_GetConversionBatch.ResultSet0** result set contains a set of conversion 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 conversion item's **JobId**.

GroupId: The conversion item's **GroupId**.

ItemId: The conversion item's **ItemId**.

InProgress: If this field is 0, then the conversion item is a not started conversion item. If this field is 1, then the conversion item is an in progress conversion item that is stale. See section 3.1.5.11 for more details.

InputFile: The conversion item's **InputFile**.

OutputFile: The conversion item's **OutputFile**.

AttemptsRemaining: The conversion item's **AttemptsRemaining**.

WorkerServerInstance: The conversion item's **WorkerServerInstance**. If the result set's **InProgress** is 0, then this field MUST be NULL.

StartTime: The conversion item's **StartTime**. If the result set's **InProgress** is 0, then this field MUST be NULL.

CreateTime: The conversion item's **CreateTime**.

2.2.5 Tables and Views

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

2.2.6 XML Structures

The namespaces, simple types, complex types, elements, and attributes that are specified in this section are used in the databaseBatchUpdate **XML element** and the databaseJobAdd 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 XML namespace prefix for each XML namespace that is used, the choice of any particular XML namespace prefix is implementation-specific and not significant for interoperability.

Prefix	Namespace URI	Reference
bu	http://schemas.microsoft.com/office/server/word/2009/08/databaseBatchUpdate	Section 2.2.6.4.1
ja	http://schemas.microsoft.com/office/server/word/2009/08/databaseJobAdd	Section 2.2.6.4.2
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1] [XMLSCHEMA2]

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 conversion items that will be updated in the database.
databaseJobAdd	Represents a conversion group that will be added to the database.

2.2.6.4.1 databaseBatchUpdate

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

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/office/server/word/2009/08/databaseBat
chUpdate" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.microsoft.com/office/server/word/2009/08/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 string that contains a long that represents the conversion item's **JobId**. Used with the **group** and **id** attributes to identify a conversion item.

batch.start.item.group: A string that contains a short that represents the conversion item's **GroupId**.

batch.start.item.id: A string that contains an integer that represents a conversion item's **ItemId**.

batch.start.item.wsi: A string that contains a uniqueidentifier that represents the new value for the conversion item's **WorkerServerInstance**.

batch.failed.item.job: A string that contains a long that represents the conversion item's **JobId**. Used with the **group** and **id** attributes to identify a conversion item.

batch.failed.item.group: A string that contains a short that represents the conversion item's **GroupId**.

batch.failed.item.id: A string that contains an integer that represents a conversion item's **ItemId**.

batch.failed.item.error: A string that contains an integer that represents the new value for the conversion item's **ErrorCode**.

Example:

```
<batch xmlns="http://schemas.microsoft.com/office/server/word/2009/08/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 conversion group that will be added to the database.

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/office/server/word/2009/08/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="optional" />
     </xs:complexType>
    </xs:element>
```

```
    </xs:sequence>
   </xs:complexType>
  </xs:element>
 </xs:schema>
```

Group.item.id: A string that contains an integer that represents the conversion item's **ItemId**.

Group.item.in: A string that represents the conversion item's **InputFile**.

Group.item.out: A string that represents the conversion item's **OutputFile**.

Example:

```
<group xmlns="http://schemas.microsoft.com/office/server/word/2009/08/databaseJobAdd">
 <item id="1" in="Aenean%20nec.docx" out="Aenean%20nec.pdf" />
 <item id="2" in="Fusce%20aliquet.docx" out="Fusce%20aliquet.pdf" />
 <item id="3" in="Lorem%20ipsum.docx" out="Lorem%20ipsum.xps" />
 <item id="4" in="Nunc%20viverra.docx" out="Nunc%20viverra.xps" />
 <item id="5" in="Pellentesque.docx" out="Pellentesque.xps" />
</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 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 conversion item represents the atomic units of file conversion. A conversion item contains information about converting a single file. A conversion item is uniquely identified by the **JobId**, **GroupId**, and **ItemId**. The information associated with a conversion item is specified as follows:

- **JobId**: The bigint that identifies a conversion job to which the conversion item belongs. This MUST NOT be NULL.
- **ItemId**: The integer that identifies a conversion item within the conversion job. Two conversion items within a conversion group MUST NOT have the same **ItemId**. This MUST NOT be NULL.
- **GroupId**: The smallint that identifies to which conversion group in the conversion job that this conversion item belongs. This MUST NOT be NULL.
- **StartTime**: The datetime when the protocol client reported that it began converting the file specified by **InputFile**. This value MUST be in **UTC**.
- **StopTime**: The datetime when the protocol client reported that it had finished converting the file specified by **InputFile**. This value MUST be in UTC.
- **AttemptsRemaining**: The tinyint specifying the number of remaining attempts the conversion item has. The meaning of the values in this field is defined by the protocol client. This MUST NOT be NULL.
- **InputFile**: The string that represents the location of the source file. This MUST NOT be NULL.
- **OutputFile**: The string that represents the location of the destination file indicating where the converted file will be placed.
- **WorkerServerInstance**: The uniqueidentifier of the worker server instance handling the conversion of this conversion item. The meaning of the values in this field is defined by the protocol client.
- **ErrorCode**: The 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.
- **Reserved**: This MUST be NULL.

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

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

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

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

- **JobId**: The uniqueidentifier that identifies a conversion job. This MUST NOT be NULL.
- **Settings**: The xml containing settings to be applied across all conversion items for the conversion job. The meaning of the values in this field is defined by the protocol client.
- **CreateTime**: The datetime that specifies when the conversion job was created. This value MUST be in UTC. This MUST NOT be NULL.
- **CancelTime**: The datetime that specifies when the conversion job was canceled. This value MUST be in UTC.
- **Submitted**: The bit that specifies if the conversion job has been submitted. Conversion items that belong in unsubmitted conversion jobs will not be converted. This MUST NOT be NULL.
- **PartitionId**: The uniqueidentifier of the partition on which the conversion job runs. The meaning of the values in this field is defined by the protocol client.
- **UserTokenHeader**: A varbinary(32). See [MS-WORDSWCF] section 2.2.3.1 for more details about **UserTokenHeader**, **UserTokenSid**, and **UserTokenGroups**. The meaning of values in this field is defined by the protocol client. If this is NULL, then **UserTokenSid** and **UserTokenGroups** MUST also be NULL. If this is NOT NULL, then **UserTokenSid** and **UserTokenGroups** MUST also be NOT NULL.
- **UserTokenSid**: A varbinary(max). Used along with **UserTokenGroups** to check user credentials. The meaning of values in this field is defined by the protocol client.
- **UserTokenGroups**: A varbinary(max). The meaning of values in this field is defined by the protocol client.
- **Name**: The nvarchar(max) 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 section specifies the following stored procedures:

Procedure Name	Description
proc_UpdateSucceededItem	Updates a conversion item that has successfully finished.
proc_UpdateFailedItem	Updates a conversion item that has failed.
proc_UpdateConversionBatch	Update multiple conversion items.
proc_SubmitJob	Marks a conversion job as entirely submitted so that the individual conversion items can begin conversion.
proc_JobsExpire	Delete conversion jobs.
proc_HasActiveJobs	Returns whether or not there are any active conversion jobs.
proc_GetJobStatus	Returns a result set specifying the status of the conversion job.
proc_GetJobs	Gets a list of conversion jobs.
proc_GetItems	Gets a result set of conversion items in a conversion job.
proc_GetGroups	Gets a result set of conversion groups of a conversion job.
proc_GetConversionBatch	Retrieves a set of conversion items that are not started or are already in progress but are stale.
proc_CancelJob	Cancels a conversion job.
proc_CancelAllActiveJobs	Cancels all conversion jobs that are active or unsubmitted.
proc_AddJob	Adds a new conversion job to the database.
proc_AddGroup	Adds a new conversion group and the conversion items that compose that conversion group to the database.

The **T-SQL** syntax for each stored procedure and result set and the variables of which they are composed is 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_UpdateSucceededItem

The **proc_UpdateSucceededItem** stored procedure is called to update a conversion item that has successfully finished. This stored procedure MUST do the following:

- The conversion item's **StopTime** is set to the current UTC date.
- The conversion item's **ErrorCode** is set to NULL.

- The conversion item's **WorkerServerInstance** is set to NULL.
- The conversion item's **Reserved** is set to **@Reserved**.

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

```
PROCEDURE proc_UpdateSucceededItem (
@JobId bigint
,@GroupId smallint
,@ItemId int
,@Reserved varbinary(max) = null
);
```

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

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

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

@Reserved: The new value for the conversion item's **Reserved** field.

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

Result Sets: MUST NOT return any result sets.

3.1.5.2 proc_UpdateFailedItem

The **proc_UpdateFailedItem** stored procedure is called to update a conversion item that has failed.

If **@NoRetry** is 0 or the conversion item's **AttemptsRemaining** is greater than 0, then the conversion 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 conversion item is updated as follows:

- The **StopTime** is set to the current UTC date.
- The **ErrorCode** is set to **@ErrorCode**.
- The **Reserved** is set to **@Reserved**.
- The **WorkerServerInstance** is set to NULL.
- The **AttemptsRemaining** is set to 0.

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

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

```
,@Reserved varbinary(max) = null
);
```

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

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

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

@NoRetry: Indicates if the conversion item can be attempted to be converted again.

@ErrorCode: The new value for the conversion item's **ErrorCode** field.

@Reserved: The new value for the conversion item's **Reserved** field.

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

Result Sets: MUST NOT return any result sets.

3.1.5.3 proc_UpdateConversionBatch

The **proc_UpdateConversionBatch** stored procedure is called to update multiple conversion items. This stored procedure also returns a result set that contains information about the conversion jobs and conversion groups that contain conversion 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 conversion item's **JobId** equals item.job.
- The conversion item's **GroupId** equals item.group.
- The conversion item's **ItemId** equals item.id.

This stored procedure updates those conversion items as follows:

- The conversion item's **StartTime** is set to the current UTC date.
- The conversion item's **WorkerServerInstance** is set to item.wsi.
- The conversion item's **AttemptsRemaining** is decremented by 1, but cannot be less than 0.

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

- The conversion item's **JobId** equals item.job.
- The conversion item's **GroupId** equals item.group.
- The conversion item's **ItemId** equals item.id.

This stored procedure updates those conversion items as follows:

- The conversion item's **WorkerServerInstance** is set to NULL.
- The conversion item's **ErrorCode** is set to item.error.

- The conversion item's **StopTime** is set to the current UTC date.

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

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

@BatchXml: The conversion 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

3.1.5.4 proc_SubmitJob

The **proc_SubmitJob** stored procedure is called to mark a conversion job as entirely submitted so that the individual conversion items can begin conversion. Any conversion job identified by **@JobId** MUST have **Submitted** set to 1 as a result of calling this stored procedure.

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

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

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

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

Result Sets: MUST NOT return any result sets.

3.1.5.5 proc_JobsExpire

The **proc_JobsExpire** stored procedure is called to delete conversion jobs. This stored procedure deletes conversion jobs that meet all of the following:

- If **@TimeThreshold** is not NULL, then conversion jobs with a **CreateTime** less than **@TimeThreshold** are eligible. If **@TimeThreshold** is NULL, then all conversion jobs are eligible unless restricted by the other parameters.
- If **@Partitionid** is not NULL, then conversion jobs with a **PartitionId** matching **@PartitionId** are eligible. If **@PartitionId** is NULL, then all conversion jobs are eligible unless restricted by the other parameters.
- If **@JobId** is not NULL, then conversion jobs with a **JobId** matching **@JobId** are eligible. If **@JobId** is NULL, then all conversion jobs are eligible unless restricted by the other parameters.
- If **@IncludeActiveJobs** is 0, then conversion jobs with a non-NULL **CreateTime** or a non-NULL **StopTime** are eligible. If **@IncludeActiveJobs** is 1, then all conversion jobs are eligible unless restricted by the other parameters.

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

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

@TimeThreshold: The UTC date specifying which conversion jobs are eligible for deletion.

@PartitionId: The uniqueidentifier specifying which conversion jobs are eligible for deletion.

@JobId: The conversion job's **JobId** specifying which conversion jobs are eligible for deletion.

@IncludeActiveJobs: This parameter specifies whether or not active conversion jobs are eligible for deletion.

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

Result Sets: MUST NOT return any result sets.

3.1.5.6 proc_HasActiveJobs

The **proc_HasActiveJobs** stored procedure returns whether or not there are any active conversion jobs. An active conversion job is a conversion job that has the following properties:

- **Submitted** is 1.
- **CancelTime** is not NULL.
- There is at least one conversion item with a NULL **StopTime**.

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

```
PROCEDURE proc_HasActiveJobs (
);
```

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

Value	Description
0	There are no active conversion jobs.
1	There is at least one active conversion job.

Result Sets: MUST NOT return any result sets.

3.1.5.7 proc_GetJobStatus

The **proc_GetJobStatus** stored procedure returns a result set specifying the status of the conversion job.

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

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

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

@PartitionId: The **PartitionId** of the conversion job that this stored procedure will return. If this parameter is not NULL, then the conversion 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

3.1.5.8 proc_GetJobs

The **proc_GetJobs** stored procedure is called to get a list of conversion jobs.

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

```
PROCEDURE proc_GetJobs (
@PartitionId uniqueidentifier = null
,@UserTokenSid varbinary(max) = null
,@UserTokenGroups varbinary(max) = null
,@ActiveOnly bit
,@SubmittedOnly bit
);
```

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

@UserTokenSid: If this parameter and **@UserTokenGroups** are both not NULL, then any conversion job returned by this stored procedure MUST have the following:

- A **UserTokenSid** that equals **@UserTokenSid**.
- A **UserTokenGroups** that equals **@UserTokenGroups**.

@UserTokenGroups: See **@UserTokenSid**.

@ActiveOnly: If this parameter is 1, then any conversion job returned by this stored procedure MUST have the following:

- A **CancelTime** that is NULL.
- A conversion item that belongs to the conversion job with a **StopTime** that is NULL.

@SubmittedOnly: If this parameter is 1, then any conversion job returned by this stored procedure MUST have a **Submitted** that equals 1.

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

Result Sets:

This stored procedure MUST return a proc_GetJobs.ResultSet0

3.1.5.9 proc_GetItems

The **proc_GetItems** stored procedure is called to get a result set of conversion items in a conversion job.

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

```
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 conversion job. Any conversion item returned by this stored procedure MUST have a **JobId** that matches **@JobId**.

@GroupId: The **GroupId** of the conversion item. Any conversion item returned by this stored procedure MUST have a **GroupId** that matches **@GroupId**.

@PartitionId: This parameter is unused.

@NotSubmitted: If this is 0, then the result set returned by this stored procedure MUST NOT contain any conversion items that have the following:

- The conversion item's conversion job has a **Submitted** equal to 0.
- The conversion item's conversion job has a **CancelTime** that is NULL.

@NotStarted: If this is 0, then the result set returned by this stored procedure MUST NOT contain any conversion items that have the following:

- The conversion item's conversion job has a **Submitted** equal to 1.
- The conversion item's conversion job has a **CancelTime** that is NULL.
- The conversion item has a **StartTime** that is NULL.

@InProgress: If this is 0, then the result set returned by this stored procedure MUST NOT contain any conversion items that have the following:

- The conversion item's conversion job has a **Submitted** equal to 1.
- The conversion item's conversion job has a **CancelTime** that is NULL.
- The conversion item has a **StartTime** that is not NULL.
- The conversion item has a **StopTime** that is NULL.

@Succeeded: If this is 0, then the result set returned by this stored procedure MUST NOT contain any conversion items that have the following:

- The conversion item's conversion job has a **Submitted** equal to 1.
- The conversion item has a **StartTime** that is not NULL.
- The conversion item has a **StopTime** that is not NULL.
- The conversion item has an **ErrorCode** that is NULL.

@Failed: If this is 0, then the result set returned by this stored procedure MUST NOT contain any conversion items that have the following:

- The conversion item's conversion job has a **Submitted** equal to 1.
- The conversion item has a **StartTime** that is not NULL.
- The conversion item has a **StopTime** that is not NULL.
- The conversion item has an **ErrorCode** that is not NULL.

@Canceled: If this is 0, then the result set returned by this stored procedure MUST NOT contain any conversion items that have the following:

- The conversion item's conversion job has a **CancelTime** that is not NULL.
- The conversion 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_GetGroups

The **proc_GetGroups** stored procedure is called to get a result set of conversion groups of a conversion job.

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

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

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

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

3.1.5.11 proc_GetConversionBatch

The **proc_GetConversionBatch** stored procedure is called to retrieve a set of conversion items that are not started or are already in progress but are stale.

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

- The conversion item's conversion job has a **Submitted** equal to 1.
- The conversion item's conversion job has a **CancelTime** that is NULL.
- The conversion item has a **StartTime** that is NULL.
- The conversion item has a **StopTime** that is NULL.

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

- The conversion item's conversion job has a **Submitted** equal to 1.
- The conversion item's conversion job has a **CancelTime** that is NULL.
- The conversion item has a **StopTime** that is NULL.
- The conversion item has a **StartTime** that is not NULL.
- The conversion item has a **StartTime** that is less than **@InProgressThreshold**.

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

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

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

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

@InProgressThreshold: The threshold that specifies which conversion 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

3.1.5.12 proc_CancelJob

The **proc_CancelJob** stored procedure is called to cancel a conversion job. This stored procedure changes the **CancelTime** of the conversion job to be the current UTC date. The following conditions MUST be true for the conversion job to be canceled.

- The conversion job's **JobId** is equal to **@JobId**.
- The conversion job's **PartitionId** is equal to **@PartitionId**.
- The conversion job's **CancelTime** is NULL.

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

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

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

@PartitionId: The **PartitionId** of the conversion job to cancel.

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

Result Sets: MUST NOT return any result sets.

3.1.5.13 proc_CancelAllActiveJobs

The proc_**CancelAllActiveJobs** stored procedure is called to cancel all conversion jobs that are active or unsubmitted. This stored procedure updates the **CancelTime** to be the current UTC date for any conversion job that is either active or unsubmitted.

An active conversion job has the following properties:

- The conversion job's **CancelTime** is NULL.
- The conversion job contains conversion items that either have a NULL **StartTime** or a NULL **StopTime**.

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

- The conversion job's **CancelTime** is NULL.
- The conversion job's **Submitted** is 0.

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

```
PROCEDURE proc_CancelAllActiveJobs (
);
```

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

Result Sets: MUST NOT return any result sets.

3.1.5.14 proc_AddJob

The **proc_AddJob** stored procedure is called to add a new conversion job to the database.

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

```
PROCEDURE proc_AddJob (
```

```
@JobId bigint
,@UserTokenHeader varbinary(32) = null
,@UserTokenSid varbinary(max) = null
,@UserTokenGroups varbinary(max) = null
,@PartitionId uniqueidentifier = null
,@Settings nvarchar(max)
,@Name nvarchar(max) = null
);
```

@JobId: The new conversion job's **JobId**. This parameter MUST NOT be NULL.

@UserTokenHeader: The new conversion job's **UserTokenHeader**.

@UserTokenSid: The new conversion job's **UserTokenSid**.

@UserTokenGroups: The new conversion job's **UserTokenGroups**.

@PartitionId: The new conversion job's **PartitionId**.

@Settings: The new conversion job's **Settings**.

@Name: The new conversion job's **Name**.

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

Result Sets: MUST NOT return any result sets.

3.1.5.15 proc_AddGroup

The **proc_AddGroup** stored procedure is called to add a new conversion group and the conversion items that compose that conversion group to the database.

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

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

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

@GroupId: The new conversion group's **GroupId**. This parameter MUST NOT be NULL.

@InputRoot: The new conversion group's **InputRoot**.

@OutputRoot: The new conversion group's **OutputRoot**.

@MaxAttemptsCount: The initial value for **AttemptsRemaining** for each of the new conversion items.

@JobXml: The xml that specifies the new conversion items within the conversion group to be added. This parameter MUST conform to the databaseJobAdd XML schema.

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

Result Sets: 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

4.1 Canonical Example

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

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

1. -- proc_AddJob -->
2. <-- return code ignored --
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 converts the conversion items.
12. -- proc_UpdateSucceededItem -->
13. <-- return code ignored --
14. -- proc_UpdateFailedItem -->
15. <-- return code ignored --
16. The client creates a conversion job and adds it to the database on the protocol server. It does this by calling the proc_AddJob stored procedure 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',

@UserTokenHeader = 0x00000000000000000000000000000000,

@UserTokenGroups = 0x1,

@UserTokenSid = 0x1,

@PartitionId = '93572c0a-d9e1-1395-dab3-932eac7ba30c',

@Settings = '<settings/>',

@Name = 'testJob'
```

1. The protocol server returns a return code, which is ignored by the client.

2. The client then adds conversion items to the conversion job by calling the proc_AddGroup stored procedure 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 = NULL,

@OutputRoot = NULL,

@MaxAttemptsCount = '2',

@JobXml =

'<group
xmlns="http://schemas.microsoft.com/office/server/word/2009/08/databaseJobAdd">

<item id="1" in="Aenean%20nec.docx" out="Aenean%20nec.pdf" />

<item id="2" in="Fusce%20aliquet.docx" out="Fusce%20aliquet.pdf" />

</group>'
```

1. The protocol server returns a return code, which is ignored by the client.

2. The client has no more conversion items to add, so the client calls the proc_SubmitJob stored procedure 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'
```

3. The protocol server returns a return code, which is ignored by the client.

4. The client queries the protocol server for which conversion 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 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'
```

5. The protocol server returns the proc_GetConversionBatch Result Set. The result set contains the following fields:

Field Name	Row 1	Row 2

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.pdf	Fusce%20aliquet.pdf
AttemptsRemaining	2	2
WorkerServerInstance	NULL	NULL
StartTime	NULL	NULL
CreateTime	2010-01-19 14:18:53.397	2010-01-19 14:18:53.397

- The protocol client updates the conversion items with the uniqueidentifier for the **WorkerServerInstance** and also to mark the conversion items it wants to start. The client does this by calling the proc_UpdateConversionBatch stored procedure 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/word/2009/08/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>'
```

- The protocol server returns the proc_UpdateConversionBatch Result Set which contains only one row in this case because there was only one conversion job that the items all belonged to:

Field Name	Value
JobId	1
GroupId	1
InputRoot	NULL
OutputRoot	NULL
Settings	<settings/>
UserTokenHeader	0x00000000000000000000000000000000
UserTokenSid	0x01
UserTokenGroups	0x01

1. The protocol client begins file conversion with the information from steps 8 and 10. With the proc_GetConversionBatch Result Set from step 8, the client has the identity of the conversion item through the **JobId**, **GroupId**, and **ItemId** as well as the source location and destination location of the file from **InputFile** and **OutputFile**. With the proc_UpdateConversionBatch Result Set, the client has more information such as the user credentials from the **UserTokenHeader**, **UserTokenSid**, and **UserTokenGroups**, any additional conversion settings for the conversion job from **Settings**, the **JobId** and **GroupId** of the conversion items, as well

as **InputRoot** and **OutputRoot**. The conversion item identified uniquely by (**JobId** =1, **GroupId** = 1, **ItemId** = 1) finishes converting first and then some time later the conversion item (**JobId** = 1, **GroupId** = 1, **ItemId** = 2) finishes but fails to convert because the document is corrupt.

2. Upon conversion completion of the conversion item (**JobId** = 1, **ConversionId** = 1), the protocol client calls the proc_UpdateSucceededItem stored procedure because the conversion item succeeded in conversion. 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',

@Reserved = NULL
```

1. The protocol server returns a return code, which is ignored by the client.

2. Upon conversion completion of the conversion item (**JobId** =1, **ConversionId** = 2), the protocol client calls the proc_UpdateFailedItem stored procedure because the conversion item failed in conversion as a result of the file being corrupt. `@NoRetry = '1'` because the protocol client has determined that the document is corrupt and will not be attempted for conversion 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',

@Reserved = NULL,

@ErrorCode = '1',

@NoRetry = '1'
```

- 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 2010

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: Office 2010 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.

8 Index

N

O

P

R

S

T

V

X