

[MS-UPSSDSP]: User Profile Social Data 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's Open Specification Promise (available here: <http://www.microsoft.com/interop/osp>) or the Community Promise (available here: <http://www.microsoft.com/interop/cp/default.mspx>). 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 iplq@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.

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	1.05	Minor	Clarified the meaning of the technical content.
09/27/2010	1.05	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	1.06	Major	Significantly changed the technical content.
12/17/2010	1.06	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	1.6.1	Editorial	Changed language and formatting in the technical content.
06/10/2011	1.6.1	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1	Introduction	7
1.1	Glossary	7
1.2	References.....	8
1.2.1	Normative References.....	8
1.2.2	Informative References	8
1.3	Protocol Overview (Synopsis)	8
1.4	Relationship to Other Protocols.....	9
1.5	Prerequisites/Preconditions	9
1.6	Applicability Statement.....	9
1.7	Versioning and Capability Negotiation.....	9
1.8	Vendor-Extensible Fields.....	9
1.9	Standards Assignments	9
2	Messages.....	10
2.1	Transport.....	10
2.2	Common Data Types	10
2.2.1	Simple Data Types and Enumerations	10
2.2.2	Bit Fields and Flag Structures.....	10
2.2.3	Binary Structures	10
2.2.4	Result Sets	10
2.2.4.1	proc_SocialComments_GetForUrlBase.ResultSet0	10
2.2.4.2	proc_SocialComments_GetForUser.ResultSet0	11
2.2.4.3	proc_SocialComments_GetForUserWithTag.ResultSet0	11
2.2.4.4	proc_SocialRatings_GetForUser.ResultSet0	12
2.2.4.5	proc_SocialRatings_GetRatingsForUpdate.ResultSet0	12
2.2.4.6	proc_SocialTags_GetForTerms.ResultSet0	13
2.2.4.7	proc_SocialTags_GetForUser.ResultSet0	13
2.2.4.8	proc_SocialTags_GetTerms.ResultSet0	14
2.2.4.9	proc_SocialTags_GetTermsForUser.proc_SocialTags_GetTermsForUser.Default.ResultSet0	14
2.2.4.10	proc_SocialTags_GetUrlsForTerm.ResultSet0	14
2.2.4.11	proc_Social_GetUsers.ResultSet0.....	14
2.2.4.12	proc_SocialComments_GetForUrl.Default.ResultSet0	14
2.2.4.13	proc_SocialRatings_GetForUrl.Default.ResultSet0	15
2.2.4.14	proc_SocialTags_GetTermsForUrl.Default.ResultSet0	15
2.2.4.15	proc_SocialTags_GetTermsForUser.ResultSet0	16
2.2.4.16	proc_SocialComments_GetDeletes.ResultSet0	16
2.2.4.17	proc_SocialTags_GetUsersForTerm.ResultSet0	16
2.2.4.18	proc_SocialTags_GetByTime.ResultSet0	17
2.2.4.19	proc_SocialTags_GetChangedUsers.ResultSet0	17
2.2.4.20	proc_SocialTags_GetDeletes.ResultSet0	17
2.2.4.21	proc_SocialTags_GetForReplicationByTime.ResultSet0	18
2.2.4.22	proc_SocialTags_GetForReplicationByUser.ResultSet0	18
2.2.4.23	proc_SocialTags_GetUrlCountsForTerm.ResultSet0	19
2.2.4.24	proc_SocialRatings_GetByTime.ResultSet0.....	19
2.2.4.25	proc_SocialRatings_GetDeletes.ResultSet0.....	20
2.2.4.26	proc_Social_GetSecurityTrimUrls.ResultSet0	20
2.2.5	Tables and Views	20
2.2.6	XML Structures	20

2.2.6.1	Namespaces	20
2.2.6.2	Simple Types	20
2.2.6.3	Complex Types.....	20
2.2.6.4	Elements	20
2.2.6.5	Attributes	21
2.2.6.6	Groups	21
2.2.6.7	Attribute Groups.....	21
3	Protocol Details.....	22
3.1	Server Details	22
3.1.1	Abstract Data Model	22
3.1.1.1	Social Tags	22
3.1.1.2	Social Comments (Notes)	22
3.1.1.3	Social Ratings	22
3.1.2	Timers	22
3.1.3	Initialization	22
3.1.4	Higher-Layer Triggered Events.....	22
3.1.5	Message Processing Events and Sequencing Rules.....	23
3.1.5.1	proc_Admin_DeletePartition	23
3.1.5.2	proc_Admin_SetupPartition	23
3.1.5.3	proc_SocialComments_Add	23
3.1.5.4	proc_SocialComments_Count.....	25
3.1.5.5	proc_SocialComments_Delete	25
3.1.5.6	proc_SocialComments_Get	26
3.1.5.7	proc_SocialComments_GetForUrl.....	27
3.1.5.8	proc_SocialComments_GetForUser	28
3.1.5.9	proc_SocialComments_GetForUserWithTag	29
3.1.5.10	proc_SocialComments_Update	29
3.1.5.11	proc_SocialRatings_Add.....	30
3.1.5.12	proc_SocialRatings_Count.....	31
3.1.5.13	proc_SocialRatings_Delete	32
3.1.5.14	proc_SocialRatings_DeleteForUser	32
3.1.5.15	proc_SocialRatings_Get	33
3.1.5.16	proc_SocialRatings_GetAverage.....	33
3.1.5.17	proc_SocialRatings_GetForUrl.....	35
3.1.5.18	proc_SocialRatings_GetForUser	35
3.1.5.19	proc_SocialRatings_GetLastAverageSyncTime	36
3.1.5.20	proc_SocialRatings_GetRatingsForUpdate	36
3.1.5.21	proc_SocialRatings_SetLastAverageSyncTime.....	37
3.1.5.22	proc_SocialTags_Add	37
3.1.5.23	proc_SocialTags_Count.....	38
3.1.5.24	proc_SocialTags_Delete	39
3.1.5.25	proc_SocialTags_DeleteForUser	40
3.1.5.26	proc_SocialTags_GetForTerms.....	40
3.1.5.27	proc_SocialTags_GetForUrl	41
3.1.5.28	proc_SocialTags_GetForUser	42
3.1.5.29	proc_SocialTags_GetTerms	43
3.1.5.30	proc_SocialTags_GetTermsForUrl.....	43
3.1.5.31	proc_SocialTags_GetTermsForUser	44
3.1.5.32	proc_SocialTags_GetUrlsForTerm.....	45
3.1.5.33	proc_SocialTags_Update	46
3.1.5.34	proc_Social_GetUsers.....	47
3.1.5.35	proc_SocialComments_GetByTime	47

3.1.5.36	proc_SocialComments_GetDeletes	48
3.1.5.37	proc_SocialComments_GetForReplicationByTime	48
3.1.5.38	proc_SocialComments_GetForReplicationByUser	49
3.1.5.39	proc_SocialComments_GetForUrlByTime	49
3.1.5.40	proc_SocialComments_ReplicateForDelete	50
3.1.5.41	proc_SocialComments_ReplicateForLive	51
3.1.5.42	proc_SocialRatings_RefreshAverages	52
3.1.5.43	proc_SocialTags_GetUsersForTerm	52
3.1.5.44	proc_SocialTags_MergeTerms	53
3.1.5.45	proc_SocialTags_RefreshTagClouds	53
3.1.5.46	proc_SocialTags_ReplicateForDelete	54
3.1.5.47	proc_SocialRatings_ReplicateForLive	55
3.1.5.48	proc_SocialTags_ReplicateForLive	55
3.1.5.49	proc_SocialTags_GetByTime	56
3.1.5.50	proc_SocialTags_GetChangedUsers	57
3.1.5.51	proc_SocialTags_GetDeletes	58
3.1.5.52	proc_SocialTags_GetForReplicationByTime	58
3.1.5.53	proc_SocialTags_GetForReplicationByUser	59
3.1.5.54	proc_SocialTags_GetForUrlByUser	59
3.1.5.55	proc_SocialTags_GetForUrlByTime	60
3.1.5.56	proc_SocialTags_GetTermsForUrlBase	61
3.1.5.57	proc_SocialTags_GetUrlCountsForTerm	62
3.1.5.58	proc_SocialRatings_GetByTime	62
3.1.5.59	proc_SocialRatings_GetDeletes	63
3.1.5.60	proc_SocialRatings_GetForReplicationByTime	63
3.1.5.61	proc_SocialRatings_GetForReplicationByUser	64
3.1.5.62	proc_SocialRatings_ReplicateForDelete	64
3.1.5.63	proc_Admin_RemoveSocialItems	65
3.1.5.64	proc_Admin_CompactChangeLogs	66
3.1.5.65	proc_SocialData_BulkDeleteForUser	66
3.1.5.66	proc_Social_SetSecurityTrimmerEnabled	67
3.1.5.67	proc_Social_RemoveSecurityTrimUrl	67
3.1.5.68	proc_Social_GetSecurityTrimUrls	68
3.1.5.69	proc_Social_GetSecurityTrimmerEnabled	68
3.1.5.70	proc_Social_AddSecurityTrimUrl	69
3.1.5.71	proc_SocialData_MergeSocialNotes	69
3.1.5.72	proc_SocialData_MigrateSocialDataUrl	70
3.1.6	Timer Events	70
3.1.7	Other Local Events	70
3.2	Client Details	70
3.2.1	Abstract Data Model	71
3.2.2	Timers	71
3.2.3	Initialization	71
3.2.4	Higher-Layer Triggered Events	71
3.2.5	Message Processing Events and Sequencing Rules	71
3.2.6	Timer Events	71
3.2.7	Other Local Events	71
4	Protocol Examples	72
4.1	Add a Social Tag to a URI	72
4.2	Add a Social Comment to a URI	74
4.3	Delete a Social Tag from a URI	76
4.4	Mark the Social Tags on a URI as Private	76

4.5	Delete a Social Comment from a URI	76
4.6	Add a Social Rating to a URI	77
4.7	Replicate Social Data.....	78
5	Security	83
5.1	Security Considerations for Implementers.....	83
5.2	Index of Security Parameters	83
6	Appendix A: Product Behavior	84
7	Change Tracking.....	85
8	Index	86

1 Introduction

This document specifies the User Profile Social Data Stored Procedure Protocol. This protocol allows clients to perform create, read, update, and delete operations on social metadata stored in a social data store on a site.

1.1 Glossary

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

Coordinated Universal Time (UTC)
GUID

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

back-end database server
base URI
content database
current user
document library
farm
list item
locale
request identifier
result set
return code
security trimming
Simple Object Access Protocol (SOAP)
site collection
social comment
social data
social rating
social tag
stored procedure
tenant
term
term store
Transact-Structured Query Language (T-SQL)
Uniform Resource Identifier (URI)
Uniform Resource Locator (URL)
user profile
user profile record identifier
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

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.

[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-SQL] 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://msdn.microsoft.com/en-us/library/dd631854\(v=SQL.10\).aspx](http://msdn.microsoft.com/en-us/library/dd631854(v=SQL.10).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>

1.2.2 Informative References

[MS-EMMSTORE] Microsoft Corporation, "[Enterprise Metadata Service Database Schema Specification](#)"

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

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

[MS-UPSPROF] Microsoft Corporation, "[User Profile Stored Procedures Protocol Specification](#)"

1.3 Protocol Overview (Synopsis)

This protocol provides a way for a protocol client to interact with the **social data** store. The social metadata store holds various attribute objects that represent information about documents and web pages that can be used across **site collections**. This protocol provides a way for the client to retrieve this information, write new information, remove information, or update existing information for each type of social data.

The social data store contains information about a document or web page that can make it easier to find and promote content. Social data is tied to individual users identified in the **user profile** store. There are three types social metadata: **social tags**, **social comments** or Notes and **social ratings**. Social Tags are descriptive keyword **terms** applied to a **URI**. Social Tags can be visible to all users or marked as private, meaning only the person who created the tag or an administrator can view it. Social comments or Notes are longer form blocks of text that allow comments or short announcements about a URI. Notes are viewable by all users. Social ratings are a judgment of 1 to 5 stars of the quality of a document in a **document library** or a **list item**. The protocol allows the protocol client to create tags, notes and ratings and set metadata on each social metadata item. This protocol facilitates the protocol clients to add, remove and update existing tags, notes and ratings, as well as query the protocol server for a list of all tags, notes, and ratings with metadata.

1.4 Relationship to Other Protocols

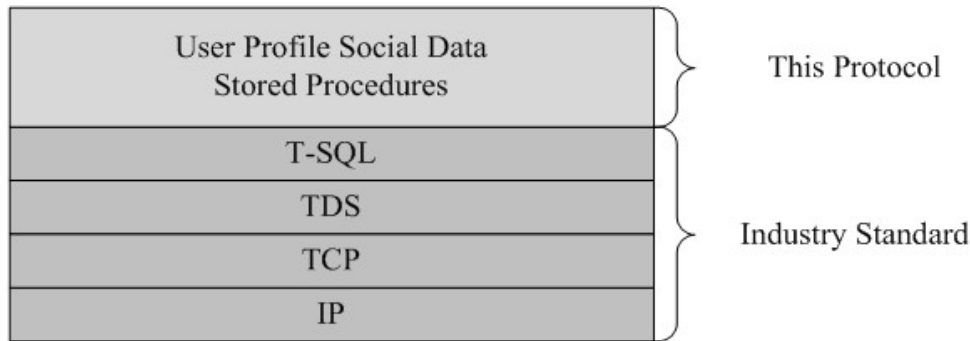


Figure 1: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

The operations described by the protocol operate between a client and a **back-end database server** on which the databases are stored. The client is expected to know the location and connection information for the databases.

This protocol requires that the protocol client has appropriate permissions to call the stored procedures stored on the back-end database server.

1.6 Applicability Statement

This protocol was designed with the intention of supporting a scale point of approximately 2 million user profiles on average of 500 social tags, 300 notes and 300 ratings.

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- **Supported Transports:** This protocol can be implemented by using transports that support sending **SOAP** (Simple Object Access Protocol) messages, as specified in section [2.1](#).
- **Protocol Versions:** This protocol is not versioned.

Capability Negotiation: This protocol does not support version negotiation.

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 the stored procedures, query SQL tables, get **return codes**, and return **result sets**.

2.2 Common Data Types

This section contains common definitions used by this protocol.

2.2.1 Simple Data Types and Enumerations

None.

2.2.2 Bit Fields and Flag Structures

None.

2.2.3 Binary Structures

None.

2.2.4 Result Sets

2.2.4.1 `proc_SocialComments_GetForUrlBase.ResultSet0`

The **`proc_SocialComments_GetForUrlBase.ResultSet0`** result set returns the social comment data corresponding to a set of input parameters.

```
CommentID bigint,  
User_RecordId bigint,  
Url nvarchar(2084),  
LastModifiedTime datetime,  
Comment nvarchar(4000),  
IsHighPriority bit,  
Title nvarchar(500),
```

CommentID: Specifies the identifier of the social comment. This value MUST NOT be NULL.

User_RecordId: Specifies the record identifier for the user profile who added the social comment. This value MUST NOT be NULL.

Url: Specifies the URI associated with the social comment. This value MUST NOT be NULL or empty.

LastModifiedTime: Specifies a timestamp in **UTC** format when the social comment was last modified. This value MUST NOT be NULL.

Comment: Specifies the text of the social comment.

IsHighPriority: A bit flag specifying if the social comment is of high priority. If the value is 1, the priority is high. Otherwise, the priority is normal. This value MUST NOT be NULL.

Title: Specifies the title of the URI for which the social comment was made at the time of last modification.

2.2.4.2 **proc_SocialComments_GetForUser.ResultSet0**

The **proc_SocialComments_GetForUser.ResultSet0** result set returns the social comment data corresponding to a set of input parameters.

```
CommentID bigint,  
Url nvarchar(2084),  
LastModifiedTime datetime,  
Comment nvarchar(4000),  
IsHighPriority bit,  
Title nvarchar(500),
```

CommentID: Specifies the identifier of the social comment. This value MUST NOT be NULL.

Url: Specifies the URI associated with the social comment. This value MUST NOT be NULL.

LastModifiedTime: Specifies a timestamp in UTC format when the social comment was last modified. This value MUST NOT be NULL.

Comment: Specifies the text of the social comment.

IsHighPriority: A bit flag specifying if the social comment is of high priority. If the value is 1, the priority is high. Otherwise, the priority is normal. This value MUST NOT be NULL.

Title: Specifies the title of the URI for which the social comment was made at the time of last modification.

2.2.4.3 **proc_SocialComments_GetForUserWithTag.ResultSet0**

The **proc_SocialComments_GetForUserWithTag.ResultSet0** result set returns the social comment data corresponding to a set of input parameters. The result set MUST be sorted on the **LastModifiedTime** column in descending order.

```
CommentID bigint,  
User_RecordID bigint,  
LastModifiedTime datetime,  
Comment nvarchar(4000),  
IsHighPriority bit,  
Title nvarchar(500),  
Url nvarchar(2084),
```

CommentID: Specifies the identifier of the social comment. This value MUST NOT be NULL.

User_RecordID: Specifies the record identifier for the user profile who added the social comment. This value MUST NOT be NULL.

LastModifiedTime: Specifies a timestamp in UTC format when the social comment was last modified. This value MUST NOT be NULL.

Comment: Specifies the text of the social comment.

IsHighPriority: A bit flag specifying if the social comment is of high priority. If the value is 1, the priority is high. Otherwise, the priority is normal. This value MUST NOT be NULL.

Title: Specifies the title of the URI for which the social comment was made at the time of last modification.

Url: Specifies the URI associated with the social comment.

2.2.4.4 **proc_SocialRatings_GetForUser.ResultSet0**

The **proc_SocialRatings_GetForUser.ResultSet0** result set contains information about the social ratings associated with a particular user profile.

```
Url nvarchar(2084),
Rating int,
LastModifiedTime datetime,
Title nvarchar(500),
```

Url: Specifies the URI associated with the social rating. This value MUST NOT be NULL or empty.

Rating: An integer specifying the social rating value in the range from 0 through 5. This value represents the 5-star rating system.

LastModifiedTime: Specifies a timestamp in UTC format when the rating was last modified. This value MUST NOT be NULL.

Title: Specifies the title of the URI for which the social ratings was made at the time of last modification.

2.2.4.5 **proc_SocialRatings_GetRatingsForUpdate.ResultSet0**

The **proc_SocialTags_GetRatingsForUpdate.ResultSet0** result set returns a list of average values of social ratings for a URI. The result set MUST contain exactly one row for each URI with a social rating.

```
Url nvarchar(2084),
AvgRating numeric(15,255),
RatingCount bigint,
ComputedTime datetime,
AverageID bigint,
```

Url: The URI. The value MUST NOT be NULL or empty.

AvgRating: The average value, in the range from 0.0 through 5.0, of social ratings for the URI. The value MUST NOT be NULL. This value represents the 5-star rating system.

RatingCount: The number of social ratings for the URI. The value MUST NOT be NULL.

ComputedTime: The time taken to compute the social rating average for this URI.

AverageID: The identifier of this social rating average.

2.2.4.6 **proc_SocialTags_GetForTerms.ResultSet0**

The **proc_SocialTags_GetForTerms.ResultSet0** result set returns a list of social tags. The result set MUST contain exactly one row for each social tag returned sorted by **LastModifiedTime** in descending order.

```
User_RecordID bigint,  
Url nvarchar(2084),  
TermID uniqueidentifier,  
LastModifiedTime datetime,  
Title nvarchar(500),  
IsPrivate bit,
```

User_RecordID: The record identifier of the user profile associated with the social tag. The value MUST NOT be NULL.

Url: The URI associated with the social tag. The value MUST NOT be NULL.

TermID: The identifier of the term associated with the social tag. The value MUST NOT be NULL.

LastModifiedTime: The date and time in UTC format when the social tag was last modified. The value MUST NOT be NULL.

Title: The title of the URI for which the social tag was made at the time of last modification. The value MUST NOT be NULL.

IsPrivate: A bit specifying if the social tag is private. If the value is 1 the social tag is private. If the value is 0 the social tag is public. The value MUST NOT be NULL.

2.2.4.7 **proc_SocialTags_GetForUser.ResultSet0**

The **proc_SocialTags_GetForUser.ResultSet0** result set returns a list of social tags. The result set MUST contain exactly one row for each social tag returned sorted by **LastModifiedTime** in descending order.

```
TermID uniqueidentifier,  
LastModifiedTime datetime,  
Title nvarchar(500),  
IsPrivate bit,  
Url nvarchar(2084),
```

TermID: The identifier of the term associated with the social tag. The value MUST NOT be NULL.

LastModifiedTime: The date and time in UTC format when the social tag was last modified. The value MUST NOT be NULL.

Title: The current title of the URI.

IsPrivate: A bit specifying if the social tag is private or public. If the value is 1 the social tag is private. If the value is 0 the social tag is public. The value MUST NOT be NULL.

Url: The URI associated with the social tag. The value MUST NOT be NULL.

2.2.4.8 **proc_SocialTags_GetTerms.ResultSet0**

The **proc_SocialTags_GetTerms.ResultSet0** result set returns a list of terms. The result set MUST contain exactly one row for each term returned sorted by **TermCount** in descending order.

```
TermID uniqueidentifier,  
TermCount bigint,
```

TermID: The identifier of the term. The value MUST NOT be NULL.

TermCount: The number of social tags using the term. The value MUST NOT be NULL.

2.2.4.9

proc_SocialTags_GetTermsForUser.proc_SocialTags_GetTermsForUser.Default.ResultSet0

The **proc_SocialTags_GetTermsForUser.ResultSet0** result set returns a list of terms. The result set MUST contain exactly one row for each term returned.

```
TermID uniqueidentifier,  
TermCount int,
```

TermID: The identifier of the term. The value MUST NOT be NULL.

TermCount: The number of social tags using the term. The value MUST NOT be NULL.

2.2.4.10 **proc_SocialTags_GetUrlsForTerm.ResultSet0**

The **proc_SocialTags_GetUrlsForTerm.ResultSet0** result set returns a list of URIs. The result set MUST contain exactly one row for each URI returned.

```
Url nvarchar(2084),
```

Url: The URI. The value MUST NOT be NULL or empty.

2.2.4.11 **proc_Social_GetUsers.ResultSet0**

The **proc_Social_GetUsers.ResultSet0** result set returns a list of user profiles. The result set MUST contain exactly one row for each user profile returned.

```
User_RecordID bigint,  
UserID uniqueidentifier,
```

User_RecordID: The record identifier of the user profile. The value MUST NOT be NULL.

UserID: The unique identifier of the user profile. The value MUST NOT be NULL.

2.2.4.12 **proc_SocialComments_GetForUrl.Default.ResultSet0**

The **proc_SocialComments_GetForUrl.proc_SocialComments_GetForUrl.Default.ResultSet0** result set returns the social comment data corresponding to a set of input parameters.

```

CommentID bigint,
User_RecordID bigint,
LastModifiedTime datetime,
Comment nvarchar(4000),
IsHighPriority bit,
Title nvarchar(500),

```

CommentID: Specifies the identifier of the social comment. This value MUST NOT be NULL.

User_RecordID: Specifies the record identifier for the user profile who added the social comment. This value MUST NOT be NULL.

LastModifiedTime: Specifies a timestamp in UTC format when the social comment was last modified. This value MUST NOT be NULL.

Comment: Specifies the text of the social comment.

IsHighPriority: A bit flag specifying if the social comment is of high priority. If the value is 1, the priority is high. Otherwise, the priority is normal. This value MUST NOT be NULL.

Title: Specifies the title of the URI for which the social comment was made at the time of last modification.

2.2.4.13 **proc_SocialRatings_GetForUrl.Default.ResultSet0**

The **proc_SocialRatings_GetForUrl.proc_SocialRatings_GetForUrl.Default.ResultSet0** result set contains information about the social ratings associated with a particular URI.

```

User_RecordID bigint,
Rating int,
LastModifiedTime datetime,

```

User_RecordID: A **user profile record identifier** associated with the social rating. This value MUST NOT be NULL.

Rating: An integer specifying the social rating value in the range from 0 through 100.

LastModifiedTime: Specifies a timestamp in UTC format when the social rating was last modified. This value MUST NOT be NULL.

2.2.4.14 **proc_SocialTags_GetTermsForUrl.Default.ResultSet0**

The **proc_SocialTags_GetTermsForUrl.proc_SocialTags_GetTermsForUrl.Default.ResultSet0** result set returns a list of terms. The result set MUST contain exactly one row for each term returned sorted by **TermCount** in descending order.

```

TermID uniqueidentifier,
TermCount int,

```

TermID: The identifier of the term. The value MUST NOT be NULL.

TermCount: The number of social tags using the term. The value MUST NOT be NULL.

2.2.4.15 **proc_SocialTags_GetTermsForUser.ResultSet0**

The **proc_SocialTags_GetTermsForUser.proc_SocialTags_GetTermsForUser.Default.ResultSet0** result set returns a list of social tags. The result set MUST contain exactly one row for each social tag returned.

```
TermID uniqueidentifier,  
TermCount int,
```

TermID: The identifier of the term associated with the social tag. The value MUST NOT be NULL.

TermCount: The number of social tags using the term. The value MUST NOT be NULL.

2.2.4.16 **proc_SocialComments_GetDeletes.ResultSet0**

The **proc_SocialComments_GetDeletes.ResultSet0** result set returns the social comment data corresponding to a set of input parameters.

```
User_RecordID bigint,  
Url nvarchar(2084),  
LastModifiedTime datetime,  
DeletedTime datetime,
```

User_RecordID: Specifies the record identifier for the user profile who added the social comment. This value MUST NOT be NULL.

Url: Specifies the URI associated with the social comment. This value MUST NOT be NULL or empty.

LastModifiedTime: Specifies a timestamp in UTC format when the social comment was last modified. This value MUST NOT be NULL.

DeletedTime: Specifies a timestamp in UTC format when the social comment was deleted. This value MUST NOT be NULL.

2.2.4.17 **proc_SocialTags_GetUsersForTerm.ResultSet0**

The **proc_SocialTags_GetUsersForTerm.ResultSet0** result set returns the user profile data corresponding to a set of input parameters. Results MUST be sorted in descending order by the last modified time of the corresponding social tags.

```
User_RecordID bigint,  
Total bigint,  
MaxLMT datetime,
```

User_RecordID: Specifies the record identifier for the user profile who added the social comment. This value MUST NOT be NULL.

Total: This value MUST be ignored by the protocol client.

MaxLMT: This value MUST be ignored by the protocol client.

2.2.4.18 **proc_SocialTags_GetByTime.ResultSet0**

The **proc_SocialTags_GetByTime.ResultSet0** result set returns the social tag data corresponding to a set of input parameters.

```
User_RecordID bigint,  
Url nvarchar(2084),  
LastModifiedTime datetime,  
TermID uniqueidentifier,  
InputTermLabel nvarchar(255),  
Title nvarchar(500),  
IsPrivate bit,
```

User_RecordID: Specifies the record identifier for the user profile who added the social comment. This value MUST NOT be NULL.

Url: The URI. The value MUST NOT be NULL.

LastModifiedTime: The date and time in UTC format when the social tag was last modified. The value MUST NOT be NULL.

TermID: The identifier of the term. The value MUST NOT be NULL.

InputTermLabel: A string representation of the input term in the user's locale at the time of input.

Title: The current title of the URI associated with the social tag. The value MUST NOT be NULL.

IsPrivate: A bit specifying if the social tag is private. If the value is 1 the social tag is private. If the value is 0 the social tag is public. The value MUST NOT be NULL.

2.2.4.19 **proc_SocialTags_GetChangedUsers.ResultSet0**

The **proc_SocialTags_GetChangedUsers.ResultSet0** result set returns the social tag data corresponding to a set of input parameters.

```
User_RecordID bigint,
```

User_RecordID: Specifies the record identifier for the user profile who added the social comment. This value MUST NOT be NULL.

2.2.4.20 **proc_SocialTags_GetDeletes.ResultSet0**

The **proc_SocialTags_GetDeletes.ResultSet0** result set returns the social tag data corresponding to a set of input parameters.

```
User_RecordID bigint,  
Url nvarchar(2084),  
TermID uniqueidentifier,  
DeletedTime datetime,
```

User_RecordID: Specifies the record identifier for the user profile who added the social comment. This value MUST NOT be NULL.

Url: The URI. The value MUST NOT be NULL.

TermID: The identifier of the term. The value MUST NOT be NULL.

DeletedTime: The date and time in UTC format when the social tag was deleted. The value MUST NOT be NULL.

2.2.4.21 **proc_SocialTags_GetForReplicationByTime.ResultSet0**

The **proc_SocialTags_GetForReplicationByTime.ResultSet0** result set returns the social tag data corresponding to a set of input parameters.

```
User_RecordID bigint,  
Url nvarchar(2084),  
TermID uniqueidentifier,  
InputTermLabel nvarchar(255),  
Title nvarchar(500),  
IsPrivate bit,  
LastModifiedTime datetime,
```

User_RecordID: Specifies the record identifier for the user profile who added the social comment. This value MUST NOT be NULL.

Url: The URI. The value MUST NOT be NULL.

TermID: The identifier of the term. The value MUST NOT be NULL.

InputTermLabel: A string representation of the input term in the user's locale at the time of input.

Title: The current title of the URI associated with the social tag. The value MUST NOT be NULL.

IsPrivate: A bit specifying if the social tag is private. If the value is 1 the social tag is private. If the value is 0 the social tag is public. The value MUST NOT be NULL.

LastModifiedTime: The date and time in UTC format when the social tag was last modified. The value MUST NOT be NULL.

2.2.4.22 **proc_SocialTags_GetForReplicationByUser.ResultSet0**

The **proc_SocialTags_GetForReplicationByUser.ResultSet0** result set returns the social tag data corresponding to a set of input parameters.

```
User_RecordID bigint,  
Url nvarchar(2084),  
TermID uniqueidentifier,  
InputTermLabel nvarchar(255),  
Title nvarchar(500),  
IsPrivate bit,  
LastModifiedTime datetime,
```

User_RecordID: Specifies the record identifier for the user profile who added the social comment. This value MUST NOT be NULL.

Url: The URI. The value MUST NOT be NULL.

TermID: The identifier of the term. The value MUST NOT be NULL.

InputTermLabel: A string representation of the input term in the user's locale at the time of input.

Title: The current title of the URI associated with the social tag. The value MUST NOT be NULL.

IsPrivate: A bit specifying if the social tag is private. If the value is 1 the social tag is private. If the value is 0 the social tag is public. The value MUST NOT be NULL.

LastModifiedTime: The date and time in UTC format when the social tag was last modified. The value MUST NOT be NULL.

2.2.4.23 **proc_SocialTags_GetUrlCountsForTerm.ResultSet0**

The **proc_SocialTags_GetUrlCountsForTerm.ResultSet0** result set returns the social tag data corresponding to a set of input parameters.

```
Url nvarchar(2084),
Title nvarchar(500),
LastModifiedTime datetime,
Total int,
```

Url: The URI. The value MUST NOT be NULL.

Title: The current title of the URI associated with the social tag. The value MUST NOT be NULL.

LastModifiedTime: The date and time in UTC format when the social tag was last modified. The value MUST NOT be NULL.

Total: The number of times the corresponding URI has been tagged with the input term.

2.2.4.24 **proc_SocialRatings_GetByTime.ResultSet0**

The **proc_SocialRatings_GetByTime.ResultSet0** result set returns the social rating data corresponding to a set of input parameters.

```
User_RecordID bigint,
Url nvarchar(2084),
LastModifiedTime datetime,
Rating int,
Title nvarchar(500),
```

User_RecordID: Specifies the record identifier for the user profile who added the social rating. This value MUST NOT be NULL.

Url: The URI. The value MUST NOT be NULL or empty.

LastModifiedTime: Specifies a timestamp in UTC format when the rating was last modified. This value MUST NOT be NULL.

Rating: An integer specifying the social rating value in the range from 0 through 5. This value represents the 5-star rating system.

Title: Specifies the title of the URI for which the social comment was made at the time of last modification.

2.2.4.25 **proc_SocialRatings_GetDeletes.ResultSet0**

The **proc_SocialRatings_GetDeletes.ResultSet0** result set returns the social rating data corresponding to a set of input parameters.

```
User_RecordID bigint,  
Url nvarchar(2084),  
DeletedTime datetime,
```

User_RecordID: Specifies the record identifier for the user profile who added the social rating. This value MUST NOT be NULL.

Url: The URI. The value MUST NOT be NULL.

DeletedTime: Specifies a timestamp in UTC format when the rating was deleted. This value MUST NOT be NULL.

2.2.4.26 **proc_Social_GetSecurityTrimUrls.ResultSet0**

```
Url nvarchar(2084),  
AlwaysAllow bit,
```

Url: The URI. The value MUST NOT be NULL or empty.

AlwaysAllow: A bit specifying whether social data whose **Url** values are hierarchical descendents of the value in the *Url* column of the results set need to pass the **security trimming** heuristics before being included in any results set returned to a user. If the value is 1, all matching social data MUST always be included in the results set returned to a user. If the value is zero, all matching social data MUST first pass the security trimming heuristics before being returned to a user. The value MUST NOT be NULL.

2.2.5 **Tables and Views**

None.

2.2.6 **XML Structures**

No common XML Structures are defined in this protocol.

2.2.6.1 **Namespaces**

No common XML Namespaces are defined in this protocol.

2.2.6.2 **Simple Types**

No common XML Simple Types are defined in this protocol.

2.2.6.3 **Complex Types**

No common XML Complex Types are defined in this protocol.

2.2.6.4 **Elements**

No common XML Structures are defined in this protocol.

2.2.6.5 Attributes

No common XML Attributes are defined in this protocol.

2.2.6.6 Groups

No common XML Groups are defined in this protocol.

2.2.6.7 Attribute Groups

No common XML Attribute Groups are defined in this protocol.

3 Protocol Details

3.1 Server Details

This section provides information about the protocol server.

3.1.1 Abstract Data Model

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

3.1.1.1 Social Tags

The protocol enables the user to add keywords to describe documents and web pages published with a URI. The social tags are stored in the protocol server. They are not part of the document or web page. Each of the tags has an identifier and is associated with a URI. Other information stored in the server include the person applying the tag, a permission flag specifying whether the tag is visible only to the person applying the tag and the date of last modification. The tag identifier is retrieved from the **term store**.

3.1.1.2 Social Comments (Notes)

The protocol enables the user to add comment text on documents and web pages published with a URI. The social comments are stored in the protocol server. They are not part of the document or web page. Each of the comments has an identifier and is associated with a URI. Other information stored in the server include the author of the comment and the date of last modification.

3.1.1.3 Social Ratings

The protocol enables the user to rate a document or list item stored on the server from 1 through 5 stars. The social ratings are stored in the protocol server. They are not part of the document or list item. Ratings are aggregated on a per item basis. The number of people who have rated the item and the average rating are periodically calculated and stored in the **content database**. Each of the ratings is associated with a URI. Other information stored in the server include the author of the rating and the date of last modification.

3.1.2 Timers

None.

3.1.3 Initialization

A connection that uses the underlying protocol layers that are specified in section [1.4](#) MUST be established before using this protocol, as specified in [\[MS-TDS\]](#).

3.1.4 Higher-Layer Triggered Events

None.

3.1.5 Message Processing Events and Sequencing Rules

3.1.5.1 proc_Admin_DeletePartition

The **proc_Admin_DeletePartition** stored procedure is used to remove the data of a specific **Tenant** in the store and to disable the store for use by that **Tenant**.

```
PROCEDURE proc_Admin_DeletePartition (  
    @partitionID uniqueidentifier  
    ,@correlationId uniqueidentifier = null  
);
```

@partitionID: A **GUID** used to filter the current request. This value MUST NOT be NULL or empty.

@correlationId: The optional **request identifier** for the current request.

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

Value	Description
1	The specified Tenant does not exist.
0	Successful execution.

Result Sets: MUST NOT return any result sets.

3.1.5.2 proc_Admin_SetupPartition

The **proc_Admin_SetupPartition** stored procedure is used to initialize the store for use by a specific Tenant.

```
PROCEDURE proc_Admin_SetupPartition (  
    @partitionID uniqueidentifier  
    ,@correlationId uniqueidentifier = null  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@correlationId: The optional request identifier for the current request.

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

Value	Description
1	The specified Tenant already exists.
0	Successful execution.

Result Sets: MUST NOT return any result sets.

3.1.5.3 proc_SocialComments_Add

The **proc_SocialComments_Add** stored procedure is called to add a new social comment for a particular URI.

```

PROCEDURE proc_SocialComments_Add (
  @partitionID uniqueidentifier
  ,@user_recordID bigint
  ,@userID uniqueidentifier
  ,@url nvarchar(2084)
  ,@comment nvarchar(4000)
  ,@isHighPriority bit
  ,@title nvarchar(500) = null
  ,@commentID bigint OUTPUT
  ,@lastModifiedTime datetime OUTPUT
  ,@correlationId uniqueidentifier = null
);

```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@user_recordID: A record identifier of the user profile associated with the social comment. This parameter MUST NOT be NULL.

@userID: A GUID for the user profile associated with the social comment. This parameter MUST NOT be NULL or empty.

@url: The URI for which the social comment has to be added. This parameter MUST be a valid URI.

@comment: Specifies the content of the social comment.

@isHighPriority: A bit flag specifying the priority of the social comment. If the parameter is "1", priority MUST be considered as high. If the parameter is zero ("0"), priority MUST be considered as normal. This parameter MUST NOT be NULL.

@title: The current title of the URI specified by **@url**.

@commentID: Output parameter indicating the identifier of the social comment just added.

Value	Description
scope_identity	The last IDENTITY value inserted into an IDENTITY column in the same scope. The IDENTITY value represents the new comment item.

@lastModifiedTime: Output parameter indicating the time when the social comment was added in UTC format.

Value	Description
getutcdate	Returns the current database system timestamp as a datetime value.

@correlationId: The optional request identifier for the current request.

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

Value	Description
1	Failure: Invalid input or internal error.
0	Successful execution.

Result Sets: MUST NOT return any result sets.

3.1.5.4 proc_SocialComments_Count

The **proc_SocialComments_Count** stored procedure is called to retrieve the count of the comments made for a given URI or user profile or both.

```
PROCEDURE proc_SocialComments_Count (  
    @partitionID uniqueidentifier  
    ,@user_recordID bigint = null  
    ,@url nvarchar(2084) = null  
    ,@count int OUTPUT  
    ,@correlationId uniqueidentifier = null  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@user_recordID: A record identifier of the user profile. If this parameter is NULL, it MUST be ignored. This parameter MUST NOT be NULL if **@url** is NULL.

@url: The URI for which the social comments have to be found. If this parameter is NULL, it MUST be ignored. If **@user_recordID** is NULL, this parameter MUST be a valid URI.

@count: Output parameter indicating the number of social comments corresponding to the specified input parameters.

Value	Description
0	An error occurred or no social comment found.
SocialComments.count	The number of social comments found.

@correlationId: The optional request identifier for the current request.

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

Value	Description
1	Failure: Invalid input or internal error.
0	Successful execution.

Result Sets: MUST NOT return any result sets.

3.1.5.5 proc_SocialComments_Delete

The **proc_SocialComments_Delete** stored procedure is called to delete a specified social comment. It can also be called to delete all social comments for a specified set of URI, user profile record identifier, and modified date.

```
PROCEDURE proc_SocialComments_Delete (  
    @partitionID uniqueidentifier  
    ,@commentID bigint = null  
    ,@user_recordID bigint = null  
    ,@url nvarchar(2084) = null  
    ,@lastModifiedTime datetime = null  
    ,@correlationId uniqueidentifier = null
```

);

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@commentID: Identifier of the social comment. If **@commentID** is NULL, **@user_RecordID**, **@url**, and **@lastModifiedTime** MUST NOT be NULL.

@user_recordID: A record identifier of the user profile associated with the social comment. If the parameter is NOT NULL and **@commentID** is NOT NULL, it MUST be equal to the record identifier of the user profile who added the comment for the stored procedure to succeed.

@url: The URI for which the social comment has to be deleted. If this parameter is NOT NULL, it MUST be a valid URI, and **@user_RecordID** and **@lastModifiedTime** MUST NOT be NULL. This parameter MUST be NULL if **@commentID** is NOT NULL.

@lastModifiedTime: A timestamp in UTC format specifying the exact last modified time of the social comment to be deleted. If this parameter is NOT NULL, **@user_RecordID** and **@url** MUST NOT be NULL. This parameter MUST be NULL if **@commentID** is NOT NULL.

@correlationId: The optional request identifier for the current request.

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

Value	Description
1	Failure: Invalid input or internal error.
0	Successful execution.

Result Sets: MUST NOT return any result sets.

3.1.5.6 proc_SocialComments_Get

The **proc_SocialComments_Get** stored procedure is called to retrieve social comments made by a particular user profile for a particular URI.

```
PROCEDURE proc_SocialComments_Get (  
    @partitionID uniqueidentifier  
    ,@user_recordID bigint = null  
    ,@url nvarchar(2084) = null  
    ,@max int = 1000  
    ,@correlationId uniqueidentifier = null  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@user_recordID: A record identifier of the user profile. This parameter MUST be ignored if its value is NULL. If this parameter is NULL, **@url** MUST NOT be NULL.

@url: A string specifying the URI. This parameter MUST be ignored if its value is NULL. If this parameter is NULL, **@user_RecordID** MUST NOT be NULL. If this parameter is NOT NULL, this parameter MUST be a valid URI.

@max: Specifies the maximum size, in rows, requested for the result set. The result set MUST NOT contain more rows than specified by **@max**. This parameter MUST be greater than or equal to 0.

@correlationId: The optional request identifier for the current request.

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

Value	Description
1	Failure: Invalid input or internal error.
0	Successful execution.

Result Sets:

For the following combination of parameters:

- **@partitionID:** select **PartitionID** from **Tenants**.
- **@url:** select **Url** from **Urls**.

This stored procedure MUST return a [proc_SocialComments_GetForUrl.Default.ResultSet0](#) result set upon successful completion. In case of failure, it MUST NOT return any result set.

3.1.5.7 proc_SocialComments_GetForUrl

The **proc_SocialComments_GetForUrl** stored procedure is called to retrieve social comments for a given URI.

```
PROCEDURE proc_SocialComments_GetForUrl (  
  @partitionID uniqueidentifier  
  ,@url nvarchar(2084)  
  ,@max int = 1000  
  ,@excludeItemsTime datetime = null  
  ,@correlationId uniqueidentifier = null  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULLI or empty.

@url: A string specifying the URI for which social comments have to be retrieved. This parameter MUST be a valid URI.

@max: Specifies the maximum size, in rows, requested for the result set. The result set MUST NOT contain more rows than specified by **@max**. This parameter MUST be greater than or equal to 0.

@excludeItemsTime: A timestamp in UTC format specifying the starting range (exclusive) for the modified date of the comments included in the result set.

@correlationId: The optional request identifier for the current request.

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

Value	Description
1	Failure: Invalid input or internal error.
0	Successful execution.

Result Sets:

For the following combination of parameters:

- **@partitionID:** Select **PartitionID** from **Tenants**.
- **@url:** Select **Url** from **Urls**.

This stored procedure MUST return a [proc_SocialComments_GetForUrl.Default.ResultSet0](#) result set upon successful completion. In case of failure, it MUST NOT return any result set.

3.1.5.8 proc_SocialComments_GetForUser

The **proc_SocialComments_GetForUser** stored procedure is called to retrieve social comments made by a given user profile.

```
PROCEDURE proc_SocialComments_GetForUser (  
    @partitionID uniqueidentifier  
    ,@user_recordID bigint  
    ,@max int = 100  
    ,@fromDate datetime = null  
    ,@toDate datetime = null  
    ,@correlationId uniqueidentifier = null  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@user_recordID: A record identifier of the user profile whose social comments have to be returned.

@max: Specifies the maximum size, in rows, requested for the result set. The result set MUST NOT contain more rows than specified by **@max**. This parameter MUST be greater than or equal to zero.

@fromDate: A timestamp in UTC format specifying the starting range (inclusive) for the modified date of the comments included in the result set. If this parameter is NULL, the protocol server MUST default this value to 1/1/2000 0:00. If this parameter is NOT NULL, **@toDate** MUST NOT be NULL.

@toDate: A timestamp in UTC format specifying the ending range (exclusive) for the modified date of the comments included in the result set. If this parameter is NULL, the protocol server MUST default this value to the next day of current system date. If this parameter is NOT NULL, **@fromDate** MUST NOT be NULL.

@correlationId: The optional request identifier for the current request.

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

Value	Description
1	Failure: Invalid input or internal error.
0	Successful execution.

Result Sets:

This stored procedure MUST return a [proc_SocialComments_GetForUser.ResultSet0](#)

3.1.5.9 **proc_SocialComments_GetForUserWithTag**

The **proc_SocialComments_GetForUserWithTag** **stored procedure** is called to return the social comments made by a particular user profile and tagged by a particular term.

```
PROCEDURE proc_SocialComments_GetForUserWithTag (
    @partitionID uniqueidentifier
    ,@user_recordID bigint
    ,@termID uniqueidentifier
    ,@correlationId uniqueidentifier = null
    ,@max int = 1000
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@user_recordID: A record identifier of the user profile. This parameter MUST NOT be NULL.

@termID: The identifier of the term. This parameter MUST NOT be NULL or empty.

@correlationId: The optional request identifier for the current request.

@max: Specifies the maximum size, in rows, requested for the result set. The result set MUST NOT contain more rows than specified by **@max**. This parameter MUST be greater than or equal to zero.

Return Values: An **integer** that MUST be zero.

Result Sets:

This stored procedure MUST return a [proc_SocialComments_GetForUserWithTag.ResultSet0](#)

3.1.5.10 **proc_SocialComments_Update**

The **proc_SocialComments_Update** stored procedure is called to update an existing social comment. The comment to be updated is identified by either **@commentID** or a combination of **@user_RecordID**, **@url**, and **@lastModifiedTime**.

```
PROCEDURE proc_SocialComments_Update (
    @partitionID uniqueidentifier
    ,@commentID bigint = null
    ,@comment nvarchar(4000) = null
    ,@isHighPriority bit
    ,@user_recordID bigint = null
    ,@url nvarchar(2084) = null
    ,@lastModifiedTime datetime OUTPUT
    ,@correlationId uniqueidentifier = null
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@commentID: Identifier of the social comment to be modified. If **@commentID** is not NULL, **@user_recordID**, **@url** MUST be NULL. If **@commentID** is NULL, **@user_recordID**, **@url** and **@lastModifiedTime** MUST NOT be NULL.

@comment: Specifies the content of the social comment modified by the user.

@isHighPriority: A bit flag specifying the priority of the comment. If the parameter is "1", priority MUST be considered as high. If the parameter is zero ("0"), priority MUST be considered as normal. This parameter MUST NOT be NULL.

@user_recordID: A record identifier of the user profile.

@url: A string specifying the URI. This parameter MUST be a valid URI.

@lastModifiedTime: A timestamp identifying the time this social comment was last modified in UTC format. As an output parameter, this parameter indicates the time when the social comment updated by this stored procedure call. The protocol server MUST set the targeted social comment's last modified time to the returned value.

Value	Description
SocialComments.getutcdate	The current database system timestamp as a datetime value.

@correlationId: The optional request identifier for the current request.

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

Value	Description
1	Failure: Invalid input or internal error.
0	Successful execution.

Result Sets: MUST NOT return any result sets.

3.1.5.11 proc_SocialRatings_Add

The **proc_SocialRatings_Add** stored procedure is called to add a new social rating. This procedure would update the rating if a rating exists for the URI **@url** by the user profile **@user_recordID**.

```
PROCEDURE proc_SocialRatings_Add (  
    @partitionID uniqueidentifier  
    ,@user_recordID bigint  
    ,@userID uniqueidentifier  
    ,@url nvarchar(2084)  
    ,@rating int  
    ,@lastModifiedTime datetime OUTPUT  
    ,@title nvarchar(500) = null  
    ,@correlationId uniqueidentifier = null  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@user_recordID: A record identifier of the user profile. This parameter MUST NOT be NULL.

@userID: An identifier for the user profile. This parameter MUST NOT be NULL or empty.

@url: A **string** specifying the URI for which the social rating has been added. This parameter MUST be a valid URI.

@rating: An **integer** value indicating the user's social rating of the URI specified by **@url**. This parameter **MUST** be greater than or equal to zero and less than or equal to 5. This value represents the 5-star rating system. This parameter **MUST NOT** be NULL.

@lastModifiedTime: Output parameter indicating the time when the rating was added in UTC format.

Value	Description
SocialRatings.getutcdate	Returns the current database system timestamp as a datetime value.

@title: The current title of the URI specified by **@url**.

@correlationId: The optional request identifier for the current request.

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

Value	Description
1	Failure: Invalid input or internal error.
0	Successful execution.

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

3.1.5.12 **proc_SocialRatings_Count**

The **proc_SocialRatings_Count** stored procedure is called to retrieve the count of social ratings for a particular URI.

```
PROCEDURE proc_SocialRatings_Count (  
    @partitionID uniqueidentifier  
    ,@url nvarchar(2084)  
    ,@count int OUTPUT  
    ,@correlationId uniqueidentifier = null  
);
```

@partitionID: A GUID used to filter the current request. This value **MUST NOT** be NULL or empty.

@url: A **string** specifying the URI. This parameter **MUST** be a valid URI.

@count: Output parameter indicating the number of social ratings corresponding to the specified input parameters.

Value	Description
0	Error occurred or no social ratings found.
SocialRatings_Averages.RatingCount	The number of social ratings corresponding to the URI specified by @url .

@correlationId: The optional request identifier for the current request.

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

Value	Description
1	Failure: Invalid input or internal error.
0	Successful execution.

Result Sets: MUST NOT return any result sets.

3.1.5.13 **proc_SocialRatings_Delete**

The **proc_SocialRatings_Delete** stored procedure is called to delete a social rating.

```
PROCEDURE proc_SocialRatings_Delete (
  @partitionID uniqueidentifier
  ,@user_recordID bigint
  ,@url nvarchar(2084)
  ,@correlationId uniqueidentifier = null
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@user_recordID: A record identifier of the user profile whose social rating has to be deleted. This parameter MUST NOT be NULL.

@url: A **string** specifying the URI whose social rating has to be deleted. This parameter MUST be a valid URI.

@correlationId: The optional request identifier for the current request.

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

Value	Description
1	Failure: Invalid input or internal error.
0	Successful execution.

Result Sets: MUST NOT return any result sets.

3.1.5.14 **proc_SocialRatings_DeleteForUser**

The **proc_SocialRatings_DeleteForUser** stored procedure is called to delete all the social ratings added by the specified user profile.

```
PROCEDURE proc_SocialRatings_DeleteForUser (
  @partitionID uniqueidentifier
  ,@user_recordID bigint
  ,@correlationId uniqueidentifier = null
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@user_recordID: A record identifier of the user profile whose social ratings have to be deleted. This parameter MUST NOT be NULL.

@correlationId: The optional request identifier for the current request.

Return Values: An **integer** that MUST be zero.

Result Sets: MUST NOT return any result sets.

3.1.5.15 **proc_SocialRatings_Get**

The **proc_SocialRatings_Get** stored procedure is called to retrieve the ratings given by a user profile for a particular URI.

```
PROCEDURE proc_SocialRatings_Get (  
    @partitionID uniqueidentifier  
    ,@user_recordID bigint  
    ,@url nvarchar(2084)  
    ,@rating int OUTPUT  
    ,@lastModifiedTime datetime OUTPUT  
    ,@title nvarchar(500) OUTPUT  
    ,@correlationId uniqueidentifier = null  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@user_recordID: A record identifier of the user profile whose social rating has to be returned. This parameter MUST NOT be NULL.

@url: A string specifying the URI whose social rating has to be returned. This parameter MUST be a valid URI.

@rating: Output parameter indicating the social rating corresponding to the specified input parameters. This value MUST be greater than or equal to zero and less than or equal to 5. This value represents the 5-star rating system.

@lastModifiedTime: Output parameter indicating the time when the social rating was last modified in UTC format.

@title: The current title of the URI specified by **@url**.

@correlationId: The optional request identifier for the current request.

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

Value	Description
1	Failure: Invalid input or internal error.
0	Successful execution.

Result Sets: MUST NOT return any result sets.

3.1.5.16 **proc_SocialRatings_GetAverage**

The **proc_SocialRatings_GetAverage** stored procedure is called to retrieve the average social rating of a given URI.

```
PROCEDURE proc_SocialRatings_GetAverage (  

```

```

@partitionID uniqueidentifier
,@url nvarchar(2084)
,@ratingAvg float OUTPUT
,@ratingCount bigint OUTPUT
,@lastModifiedTime datetime OUTPUT
,@averageID bigint OUTPUT
,@correlationId uniqueidentifier = null
);

```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@url: A **string** specifying the URI for which the average social rating has to be returned. This parameter MUST be a valid URI.

@ratingAvg: Output parameter indicating the average social rating of the URI specified by **@url**. The result MUST be greater than or equal to zero and less than or equal to 5.0, except when the item has not yet been rated, in which case it MUST be NULL. This value represents the 5-star rating system.

Value	Description
SocialRatings_Averages.AvgRating	Average social rating of the URI specified by @url .

@ratingCount: Output parameter indicating the number of social ratings corresponding to the specified input parameters.

Value	Description
SocialRatings_Averages.RatingCount	Number of social ratings corresponding to the specified input parameters.

@lastModifiedTime: Output parameter indicating the time when the social rating was last modified in UTC format.

Value	Description
SocialRatings_Averages.ComputedTime	The timestamp when the social rating was last computed in UTC format.

@averageID: Output parameter indicating the identifier of the social rating average.

Value	Description
SocialRatings_Averages.UrlID	An identifier of the URI corresponding to @url .

@correlationId: The optional request identifier for the current request.

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

Value	Description
1	Failure: Invalid input or internal error.
0	Successful execution.

Result Sets: MUST NOT return any result sets.

3.1.5.17 **proc_SocialRatings_GetForUrl**

The **proc_SocialRatings_GetForUrl** is called to retrieve all the social ratings corresponding to a particular URI.

```
PROCEDURE proc_SocialRatings_GetForUrl (  
    @partitionID uniqueidentifier  
    ,@url nvarchar(2084)  
    ,@correlationId uniqueidentifier = null  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@url: A string specifying the URI whose social ratings has to be returned. This parameter MUST be a valid URI.

@correlationId: The optional request identifier for the current request.

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

Value	Description
1	Failure: Invalid input or internal error.
0	Successful execution.

Result Sets:

For the following combination of parameters:

- **@partitionID:** Select **PartitionID** from **Tenants**.
- **@url:** Select **Url** from **Urls**.

This stored procedure MUST return a [proc_SocialRatings_GetForUrl.Default.ResultSet0](#) result set upon successful completion. In case of failure, it MUST NOT return any result set.

3.1.5.18 **proc_SocialRatings_GetForUser**

The **proc_SocialRatings_GetForUser** stored procedure is called to retrieve all the social ratings associated with a particular user profile

```
PROCEDURE proc_SocialRatings_GetForUser (  
    @partitionID uniqueidentifier  
    ,@user_recordID bigint  
    ,@correlationId uniqueidentifier = null  
    ,@top int = 1000  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@user_recordID: A record identifier of the user profile whose social ratings have to be retrieved. This parameter MUST NOT be NULL.

@correlationId: The optional request identifier for the current request.

@top: Specifies the maximum size, in rows, requested for the result set. The result set MUST NOT contain more rows than specified by **@top**. This value MUST NOT be NULL or negative.

Return Values: An **integer** that MUST be zero.

Result Sets:

This stored procedure MUST return a [proc_SocialRatings_GetForUser.ResultSet0](#)

3.1.5.19 **proc_SocialRatings_GetLastAverageSyncTime**

The **proc_SocialRatings_GetLastAverageSyncTime** stored procedure is invoked to get the most recent time that the average values of social ratings were synchronized from the social metadata store to the schemas of the corresponding list items in a **farm** specified by the **@farmID** parameter.

```
PROCEDURE proc_SocialRatings_GetLastAverageSyncTime (  
    @partitionID uniqueidentifier  
    ,@farmID uniqueidentifier  
    ,@lastSyncTime datetime OUTPUT  
    ,@lastAverageID bigint OUTPUT  
    ,@correlationId uniqueidentifier = null  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@farmID: The identifier of the farm. The value MUST NOT be NULL.

@lastSyncTime: The most recent synchronization date and time in UTC format.

@lastAverageID: The identifier of the most recently synchronized social rating average.

@correlationId: The optional request identifier for the current request.

Return Values: An **integer** that MUST be zero.

Result Sets: MUST NOT return any result sets.

3.1.5.20 **proc_SocialRatings_GetRatingsForUpdate**

The **proc_SocialRatings_GetRatingsForUpdate** stored procedure is invoked to get a set of average values of social ratings. The result set MUST be in ascending order by last modified time and **urlID**.

```
PROCEDURE proc_SocialRatings_GetRatingsForUpdate (  
    @partitionID uniqueidentifier  
    ,@lastUpdateTime datetime  
    ,@lastAverageID bigint  
    ,@batchSize int  
    ,@correlationId uniqueidentifier = null  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@lastUpdateTime: Date and time in UTC format used to filter the result set. Social rating averages modified at a time equal to this value are included in the result set only if their **urlID** is greater than the **@lastAverageID** parameter. Social rating averages modified at a time greater than this value are also included in the result set. The value MUST NOT be set to NULL.

@lastAverageID: Identifier of a social rating average used to filter the result set. Social rating averages included in the result set that were modified at a time equal to the **@lastUpdateTime** parameter MUST have an identifier greater than this value. The value MUST NOT be set to NULL.

@batchSize: The maximum number of items returned in the result set. If more results exist then they are discarded. The value MUST be greater than or equal to zero.

@correlationId: The optional request identifier for the current request.

Return Values: An **integer** that MUST be zero.

Result Sets:

This stored procedure MUST return a [proc_SocialRatings_GetRatingsForUpdate.ResultSet0](#)

3.1.5.21 proc_SocialRatings_SetLastAverageSyncTime

The **proc_SocialRatings_SetLastAverageSyncTime** stored procedure is invoked to set the most recent time that the average values of social ratings were synchronized from the social metadata store to the schemas of the corresponding list items in a farm specified by the **@farmID** parameter.

```
PROCEDURE proc_SocialRatings_SetLastAverageSyncTime (  
    @partitionID uniqueidentifier  
    ,@farmID uniqueidentifier  
    ,@lastSyncTime datetime  
    ,@lastAverageID bigint  
    ,@correlationId uniqueidentifier = null  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@farmID: The identifier of the farm. The value MUST NOT be set to NULL.

@lastSyncTime: The synchronization date and time in UTC format. The value MUST NOT be set to NULL.

@lastAverageID: Identifier of the most recently updated social rating average. The value MUST NOT be set to NULL.

@correlationId: The optional request identifier for the current request.

Return Values: An **integer** that MUST be zero.

Result Sets: MUST NOT return any result sets.

3.1.5.22 proc_SocialTags_Add

The **proc_SocialTags_Add** stored procedure is invoked to add a new social tag. If the social tag already exists, it's last modified time, term label, **isPrivate** bit and title are updated as specified in the input parameters.

```

PROCEDURE proc_SocialTags_Add (
  @partitionID uniqueidentifier
  ,@user_recordID bigint
  ,@userID uniqueidentifier
  ,@url nvarchar(2084)
  ,@termID uniqueidentifier
  ,@inputTermLabel nvarchar(255)
  ,@title nvarchar(500) = null
  ,@isPrivate bit
  ,@lastModifiedTime datetime OUTPUT
  ,@correlationId uniqueidentifier = null
);

```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@user_recordID: The record identifier of the user profile associated with the social tag. The value MUST NOT be NULL.

@userID: The unique identifier of the user profile associated with the social tag. The value MUST NOT be NULL.

@url: The URI associated with the social tag. This parameter MUST be a valid URI.

@termID: The identifier of the term associated with the social tag. The value MUST NOT be NULL.

@inputTermLabel: The label of the term identified by **@termID**.

@title: The current title of the URI specified by **@url**.

@isPrivate: A bit specifying if the social tag is private or public. If the value is "1", the social tag is private. If the value is zero ("0"), the social tag is public. The value MUST NOT be NULL.

@lastModifiedTime: Output parameter indicating the date and time when the social tag was added in UTC format.

@correlationId: The optional request identifier for the current request.

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

Value	Description
1	Failure: Invalid input or internal error.
0	Successful execution.

Result Sets: MUST NOT return any result sets.

3.1.5.23 proc_SocialTags_Count

The **proc_SocialTags_Count** stored procedure is invoked to count the number of social tags associated with a specific user profile or for a specific URI or for both.

```

PROCEDURE proc_SocialTags_Count (
  @partitionID uniqueidentifier
  ,@user_recordID bigint = null
  ,@url nvarchar(2084) = null

```

```
,@includePrivate bit
,@count int OUTPUT
,@correlationId uniqueidentifier = null
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@user_recordID: The record identifier of the user profile. If this value is NULL, every social tag associated with the URI specified by **@url** is included.

@url: The URI. If the value is NULL, every social tag associated with the user profile is included. If both **@user_recordID** and **@url** are NULL, the return value is "1", indicating input is not valid, and the output parameter **@count** is set to zero. If this parameter is not NULL, it MUST be a valid URI.

@includePrivate: A bit specifying whether to include private social tags. If the value is "1", private social tags are included. If the value is zero("0"), private social tags are not included. The value MUST NOT be NULL.

@count: An **integer** specifying the number of social tags corresponding to the input parameters.

@correlationId: The optional request identifier for the current request.

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

Value	Description
1	Failure: Invalid input or internal error.
0	Successful execution.

Result Sets: MUST NOT return any result sets.

3.1.5.24 proc_SocialTags_Delete

The **proc_SocialTags_Delete** stored procedure is invoked to delete a social tag associated with a specific user profile for a specific URI using a specific term.

```
PROCEDURE proc_SocialTags_Delete (
  @partitionID uniqueidentifier
  ,@user_recordID bigint
  ,@url nvarchar(2084)
  ,@termID uniqueidentifier = null
  ,@correlationId uniqueidentifier = null
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@user_recordID: The record identifier of the user profile. The value MUST NOT be NULL.

@url: The URI. This parameter MUST be a valid URI.

@termID: The identifier of the term. If the value is NULL, all social tags belonging to the user profile at the URI are deleted.

@correlationId: The optional request identifier for the current request.

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

Value	Description
1	Failure: Invalid input or internal error.
0	Successful execution.

Result Sets: MUST NOT return any result sets.

3.1.5.25 **proc_SocialTags_DeleteForUser**

The **proc_SocialTags_DeleteForUser** stored procedure is invoked to delete all the social tags associated with a specific user profile for a specific URI.

```
PROCEDURE proc_SocialTags_DeleteForUser (  
    @partitionID uniqueidentifier  
    ,@user_recordID bigint  
    ,@url nvarchar(2084) = null  
    ,@correlationId uniqueidentifier = null  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@user_recordID: The record identifier of the user profile. The value MUST NOT be NULL.

@url: The URI. If this value is NULL, every social tag belonging to the user profile MUST be deleted. If this parameter is not NULL, it MUST be a valid URI.

@correlationId: The optional request identifier for the current request.

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

Value	Description
1	Failure: Invalid input or internal error.
0	Successful execution.

Result Sets: MUST NOT return any result sets.

3.1.5.26 **proc_SocialTags_GetForTerms**

The **proc_SocialTags_GetForTerms** stored procedure is invoked to get the set of social tags associated with any term in a set of terms specified by the **@termIDs** parameter.

```
PROCEDURE proc_SocialTags_GetForTerms (  
    @partitionID uniqueidentifier  
    ,@viewer_user_recordID bigint = null  
    ,@includePrivate bit = 0  
    ,@termIDs varchar(max)  
    ,@top int = 1000  
    ,@correlationId uniqueidentifier = null  
);
```


@partitionID: A GUID used to filter the current request. This value MUST NOT be null or empty.

@viewer_user_recordID: The record identifier of a user profile. If the value is NOT NULL then the result set MUST include private social tags that belong to the specified user profile. If the value is NULL, no additional private comments are included.

@includePrivate: A bit specifying if the result set is to include private social tags. If the value is 1, private social tags are included. If the value is 0, private social tags are not included, except for those identified by the **@viewer_user_recordID** parameter. The value MUST NOT be set to NULL.

@termIDs: The identifiers of the terms. The value must be a space delimited list of unique identifiers, for example: "GUID1 GUID2 GUID3".

@top: Specifies the maximum size, in rows, requested for the result set. The result set MUST NOT contain more rows than specified by **@top**. This value MUST NOT be NULL or negative.

@correlationId: The optional request identifier for the current request.

Return Values: An integer that MUST be 0.

Result Sets:

This stored procedure MUST return a [proc_SocialTags_GetForTerms.ResultSet0](#)

3.1.5.27 proc_SocialTags_GetForUrl

The **proc_SocialTags_GetForUrl** stored procedure is invoked to get the set of social tags for a specific URI.

```
PROCEDURE proc_SocialTags_GetForUrl (  
    @viewer_user_recordID bigint = null  
    ,@includePrivate bit = 0  
    ,@url nvarchar(2084) = null  
    ,@partitionID uniqueidentifier  
    ,@top int = 12  
    ,@correlationId uniqueidentifier = null  
);
```

@viewer_user_recordID: The record identifier of a user profile. If the value is NOT NULL, the result set MUST include private social tags that belong to the specified user profile. If the value is NULL, no additional private comments are included.

@includePrivate: A bit specifying if the result set is to include private social tags. If the value is 1, private social tags are included. If the value is 0, private social tags are not included, except for those identified by the **@viewer_user_recordID** parameter. The value MUST NOT be set to NULL.

@url: The URI. This parameter MUST be a valid URI.

@partitionID: A GUID used to filter the current request. This value MUST NOT be null or empty.

@top: Specifies the maximum size, in rows, requested for the result set. The result set MUST NOT contain more rows than specified by **@top**. This value MUST NOT be NULL, negative, or greater than 1000.

@correlationId: The optional request identifier for the current request.

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

Value	Description
1	Failure: Invalid input or internal error.
0	Successful execution.

Result Sets:

This stored procedure MUST return a [proc_SocialTags_GetForTerms.ResultSet0](#)

3.1.5.28 proc_SocialTags_GetForUser

The **proc_SocialTags_GetForUser** stored procedure is invoked to get the set of social tags for a specific user profile.

```

PROCEDURE proc_SocialTags_GetForUser (
  @partitionID uniqueidentifier
  ,@user_recordID bigint
  ,@top int = 1000
  ,@includePrivate bit
  ,@fromDate datetime = null
  ,@toDate datetime = null
  ,@termID uniqueidentifier = null
  ,@correlationId uniqueidentifier = null
);

```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@user_recordID: The record identifier of the user profile. The value MUST NOT be NULL.

@top: Specifies the maximum size, in rows, requested for the result set. The result set MUST NOT contain more rows than specified by **@top**. This value MUST NOT be NULL or negative.

@includePrivate: A bit specifying if the result set is to include private social tags. If the value is "1", private social tags are included. If the value is zero ("0"), private social tags are not included. The value MUST NOT be NULL.

@fromDate: Date and time in UTC format used to filter the result set. If the value is not NULL, social tags included in the result set MUST have a last modified time greater than this value. If the value is NULL, the protocol server MUST NOT filter results based on last modified dates. If this value is NULL then **@toDate** MUST be set to NULL. If the value is not NULL then **@toDate** MUST NOT be NULL.

@toDate: Date and time in UTC format used to filter the result set. If the value is not NULL, social tags included in the result set MUST have a last modified time less than this value. If the value is NULL, the protocol server MUST NOT filter results based on last modified dates. If this value is NULL, **@fromDate** MUST be NULL. If the value is not NULL, **@fromDate** MUST NOT be NULL.

@termID: The identifier of the term. The value MUST NOT be NULL.

@correlationId: The optional request identifier for the current request.

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

Value	Description
1	Failure: Invalid input or internal error.
0	Successful execution.

Result Sets:

This stored procedure MUST return a [proc_SocialTags_GetForUser.ResultSet0](#)

3.1.5.29 proc_SocialTags_GetTerms

The **proc_SocialTags_GetTerms** stored procedure is invoked to get the set of terms used by social tags.

```
PROCEDURE proc_SocialTags_GetTerms (
    @partitionID uniqueidentifier
    ,@top int = 1000
    ,@correlationId uniqueidentifier = null
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@top: Specifies the maximum size, in rows, requested for the result set. The result set MUST NOT contain more rows than specified by **@top**. This value MUST NOT be NULL or negative.

@correlationId: The optional request identifier for the current request.

Return Values: An **integer** that MUST be zero.

Result Sets:

This stored procedure MUST return a [proc_SocialTags_GetTerms.ResultSet0](#)

3.1.5.30 proc_SocialTags_GetTermsForUrl

The **proc_SocialTags_GetTermsForUrl** stored procedure is invoked to get the set of terms used by social tags for a specific URI.

```
PROCEDURE proc_SocialTags_GetTermsForUrl (
    @partitionID uniqueidentifier
    ,@viewer_user_recordID bigint = null
    ,@includePrivate bit = 0
    ,@url nvarchar(2084)
    ,@top int = 100
    ,@correlationId uniqueidentifier = null
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be null or empty.

@viewer_user_recordID: The record identifier of a user profile. If the value is NOT NULL, the result set MUST include private social tags that belong to the specified user profile. If the value is NULL, no additional private comments are included.

@includePrivate: A bit specifying if the result set is to include private social tags. If the value is 1, private social tags are included. If the value is 0, private social tags are not included, except for those identified by the **@viewer_user_recordID** parameter. The value MUST NOT be set to NULL.

@url: The URI. This parameter MUST be a valid URI.

@top: Specifies the maximum size, in rows, requested for the result set. The result set MUST NOT contain more rows than specified by **@top**. This value MUST NOT be NULL, negative, or greater than 1000.

@correlationId: The optional request identifier for the current request.

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

Value	Description
1	Failure: Invalid input or internal error.
0	Successful execution.

Result Sets:

This stored procedure MUST return a [proc_SocialTags_GetTermsForUrl.Default.ResultSet0](#) upon successful completion. In case of error it MUST NOT return any result set.

3.1.5.31 proc_SocialTags_GetTermsForUser

The **proc_SocialTags_GetTermsForUser** stored procedure is invoked to get the set of terms used by social tags for a specific user profile.

```
PROCEDURE proc_SocialTags_GetTermsForUser (  
    @partitionID uniqueidentifier  
    ,@user_recordID bigint  
    ,@includePrivate bit  
    ,@top int = 1000  
    ,@correlationId uniqueidentifier = null  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@user_recordID: The identifier of the user profile. The value MUST NOT be NULL.

@includePrivate: A bit specifying if the result set is to include private social tags. If the value is "1", private social tags are included. If the value is zero ("0"), private social tags are not included. The value MUST NOT be NULL.

@top: Specifies the maximum size, in rows, requested for the result set. The result set MUST NOT contain more rows than specified by **@top**. This value MUST NOT be NULL, negative, or greater than 3000.

@correlationId: The optional request identifier for the current request.

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

Value	Description
1	Failure: Invalid input or internal error.
0	Successful execution.

Result Sets:

This stored procedure MUST return a

[proc_SocialTags_GetTermsForUser.proc_SocialTags_GetTermsForUser.Default.ResultSet0](#)

For the following combination of parameters:

- **@partitionID:** Select **PartitionID** from **Tenants**.
- **@user_recordID:** Select **User_RecordID** from **UserProfile_Validation**.
- **@url:** Select **Url** from **Urls**.

If the **@URI** parameter's value is a valid URI with one or more social tags, this stored procedure MUST return a [proc_SocialTags_GetTermsForUser.ResultSet0](#) upon successful completion.

If the **@URI** parameter's value is not a valid URI OR a valid URI without any social tags, this stored procedure MUST NOT return a result set.

3.1.5.32 proc_SocialTags_GetUrlsForTerm

The **proc_SocialTags_GetUrlsForTerm** stored procedure is invoked to get the set of URIs associated with social tags having a specified term and user profile combination. The term is specified using the **@termID** parameter. The user profile is specified using the **@user_recordID** parameter.

```

PROCEDURE proc_SocialTags_GetUrlsForTerm (
    @partitionID uniqueidentifier
    ,@termID uniqueidentifier
    ,@user_recordID bigint
    ,@includePrivate bit = 0
    ,@top int = 400
    ,@correlationId uniqueidentifier = null
);

```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@termID: The identifier of the term. The value MUST NOT be NULL.

@user_recordID: The unique identifier of the user profile. The value MUST NOT be NULL.

@includePrivate: A bit specifying if the result set is to include private social tags. If the value is "1", private social tags are included. If the value is zero ("0"), private social tags are not included. The value MUST NOT be NULL.

@top: Specifies the maximum size, in rows, requested for the result set. The result set MUST NOT contain more rows than specified by **@top**. This value MUST NOT be NULL, negative, or greater than 2000.

@correlationId: The optional request identifier for the current request.

Return Values: An **integer** that MUST be zero.

Result Sets:

This stored procedure MUST return a [proc_SocialTags_GetUrlsForTerm.ResultSet0](#)

3.1.5.33 proc_SocialTags_Update

The **proc_SocialTags_Update** stored procedure is invoked to update an existing social tag. The social tag is identified by its URI specified by the **@url** parameter, its term specified by the **@termID** parameter and its user profile specified by the **@user_recordID** parameter.

```
PROCEDURE proc_SocialTags_Update (  
    @partitionID uniqueidentifier  
    ,@user_recordID bigint  
    ,@url nvarchar(2084)  
    ,@termID uniqueidentifier  
    ,@title nvarchar(500) = null  
    ,@isPrivate bit  
    ,@lastModifiedTime datetime OUTPUT  
    ,@correlationId uniqueidentifier = null  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@user_recordID: The record identifier of the user profile associated with the social tag. The value MUST NOT be NULL.

@url: The URI associated with the social tag. This parameter MUST be a valid URI.

@termID: The identifier of the term used by the social tag. The value MUST NOT be NULL.

@title: The new title of the social tag. If the value is NULL, the title is not updated.

@isPrivate: A bit specifying if the social tag is private or public. If the value is "1", the social tag is private. If the value is zero ("0"), the social tag is public. The social tag is updated with this value. The value MUST NOT be NULL.

@lastModifiedTime: Output parameter indicating the date and time when the social tag was updated in UTC format.

@correlationId: The optional request identifier for the current request.

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

Value	Description
1	Failure: Invalid input or internal error.
0	Successful execution.

Result Sets: MUST NOT return any result sets.

3.1.5.34 **proc_Social_GetUsers**

The **proc_Social_GetUsers** stored procedure is called to retrieve the list of user profiles that have stored social data.

```
PROCEDURE proc_Social_GetUsers (  
    @partitionID uniqueidentifier  
    ,@top int = 1000  
    ,@correlationId uniqueidentifier = null  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@top: Specifies the maximum size, in rows, requested for the result set. The result set MUST NOT contain more rows than specified by **@top**. This value MUST NOT be NULL or negative.

@correlationId: The optional request identifier for the current request.

Return Values: An **integer** that MUST be zero.

Result Sets:

This stored procedure MUST return a [proc_Social_GetUsers.ResultSet0](#)

3.1.5.35 **proc_SocialComments_GetByTime**

The **proc_SocialComments_GetByTime** stored procedure is called to retrieve social comments made during a specific period of time.

```
PROCEDURE proc_SocialComments_GetByTime (  
    @partitionID uniqueidentifier  
    ,@startTime datetime  
    ,@endTime datetime  
    ,@max int = 1000  
    ,@correlationId uniqueidentifier = null  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@startTime: A timestamp in UTC format specifying the starting range, exclusive, for the modified date of the comments included in the result set.

@endTime: A timestamp in UTC format specifying the ending range, inclusive, for the modified date of the comments included in the result set.

@max: The maximum size, in rows, requested for the result set. The result set MUST NOT contain more rows than specified by **@max**. This parameter MUST be greater than or equal to zero.

@correlationId: The optional request identifier for the current request.

Return Values: An **integer** that MUST be zero.

Result Sets:

This stored procedure MUST return a [proc_SocialComments_GetForUrlBase.ResultSet0](#)

3.1.5.36 **proc_SocialComments_GetDeletes**

The **proc_SocialComments_GetDeletes** stored procedure is called to retrieve social comments deleted during a specific period of time.

```
PROCEDURE proc_SocialComments_GetDeletes (  
    @partitionID uniqueidentifier  
    ,@startTime datetime  
    ,@endTime datetime  
    ,@correlationId uniqueidentifier = null  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@startTime: A timestamp in UTC format specifying the starting range, inclusive, for the modified date of the comments included in the result set.

@endTime: A timestamp in UTC format specifying the ending range, exclusive, for the modified date of the comments included in the result set.

@correlationId: The optional request identifier for the current request.

Return Values: An **integer** that MUST be zero.

Result Sets:

This stored procedure MUST return a [proc_SocialComments_GetDeletes.ResultSet0](#)

3.1.5.37 **proc_SocialComments_GetForReplicationByTime**

The **proc_SocialComments_GetForReplicationByTime** stored procedure is called to retrieve social comments made during a specific period of time.

```
PROCEDURE proc_SocialComments_GetForReplicationByTime (  
    @partitionID uniqueidentifier  
    ,@startTime datetime  
    ,@endTime datetime  
    ,@correlationId uniqueidentifier = null  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@startTime: A timestamp in UTC format specifying the starting range, inclusive, for the modified date of the comments included in the result set.

@endTime: A timestamp in UTC format specifying the ending range, exclusive, for the modified date of the comments included in the result set.

@correlationId: The optional request identifier for the current request.

Return Values: An **integer** that MUST be zero.

Result Sets:

This stored procedure MUST return a [proc_SocialComments_GetForUrlBase.ResultSet0](#)

3.1.5.38 **proc_SocialComments_GetForReplicationByUser**

The **proc_SocialComments_GetForReplicationByUser** stored procedure is called to retrieve social comments made by a specific user profile.

```
PROCEDURE proc_SocialComments_GetForReplicationByUser (  
    @partitionID uniqueidentifier  
    ,@user_RecordID int  
    ,@correlationId uniqueidentifier = null  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@user_RecordID: A record identifier of the user profile associated with the social comment. This parameter MUST NOT be NULL.

@correlationId: The optional request identifier for the current request.

Return Values: An **integer** that MUST be zero.

Result Sets:

This stored procedure MUST return a [proc_SocialComments_GetForUrlBase.ResultSet0](#)

3.1.5.39 **proc_SocialComments_GetForUrlByTime**

The **proc_SocialComments_GetForUrlByTime** stored procedure is called to retrieve social comments for a specific **URL** made by a specific user profile.

```
PROCEDURE proc_SocialComments_GetForUrlByTime (  
    @url nvarchar(2084)  
    ,@partitionID uniqueidentifier  
    ,@user_recordID bigint = null  
    ,@fromDate datetime = null  
    ,@toDate datetime = null  
    ,@top int = 100  
    ,@correlationId uniqueidentifier = null  
);
```

@url: A **string** specifying the URI for which social comments have to be retrieved. This parameter MUST be a valid URI.

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@user_recordID: A record identifier of the user profile associated with the social comment. This parameter MUST NOT be NULL.

@fromDate: A timestamp in UTC format specifying the starting range, inclusive, for the modified date of the comments included in the result set.

@toDate: A timestamp in UTC format specifying the ending range, exclusive, for the modified date of the comments included in the result set.

@top: Specifies the maximum size, in rows, requested for the result set. The result set MUST NOT contain more rows than specified by **@top**. This value MUST NOT be NULL, negative, or greater than 100.

@correlationId: The optional request identifier for the current request.

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

Value	Description
1	Failure: Invalid input or internal error.
0	Successful execution.

Result Sets:

For the following combination of parameters:

- **@url:** Select **Url** from **Urls**.
- **@partitionID:** Select **PartitionID** from **Tenants**.

This stored procedure MUST return a [proc_SocialComments_GetForUrlBase.ResultSet0](#)

3.1.5.40 proc_SocialComments_ReplicateForDelete

The **proc_SocialComments_ReplicateForDelete** stored procedure is called to update a social comment with a deletion time.

```
PROCEDURE proc_SocialComments_ReplicateForDelete (  
    @partitionID uniqueidentifier  
    ,@user_recordID bigint  
    ,@url nvarchar(2084)  
    ,@lastModifiedTime datetime  
    ,@deletedTime datetime  
    ,@correlationId uniqueidentifier = null  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@user_recordID: A record identifier of the user profile associated with the social comments. This parameter MUST NOT be NULL.

@url: A string specifying the URI for which social comments have to be retrieved. This parameter MUST be a valid URI.

@lastModifiedTime: A timestamp in UTC format specifying the time when the social comment was last modified.

@deletedTime: A timestamp in UTC format specifying the time when the social comment was deleted.

@correlationId: The optional request identifier for the current request.

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

Value	Description
1	Failure: Invalid input or internal error.

Value	Description
0	Successful execution.

Result Sets: MUST NOT return any result sets.

3.1.5.41 **proc_SocialComments_ReplicateForLive**

The **proc_SocialComments_ReplicateForLive** stored procedure is called to add or update a social comment that has not been deleted.

```

PROCEDURE proc_SocialComments_ReplicateForLive (
  @partitionID uniqueidentifier
  ,@user_recordID bigint
  ,@userID uniqueidentifier
  ,@url nvarchar(2084)
  ,@comment nvarchar(4000)
  ,@isHighPriority bit
  ,@title nvarchar(500) = null
  ,@lastModifiedTime datetime
  ,@correlationId uniqueidentifier = null
);

```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@user_recordID: A record identifier of the user profile associated with the social comment. This parameter MUST NOT be NULL.

@userID: A record identifier of the user profile associated with the social comment. This parameter MUST NOT be NULL.

@url: A **string** specifying the URI for which social comments have to be retrieved. This parameter MUST be a valid URI.

@comment: The content of the social comment. This parameter MUST NOT be NULL or empty.

@isHighPriority: A bit flag specifying the priority of the social comment. If the parameter is "1", priority MUST be considered as high. If the parameter is zero ("0"), priority MUST be considered as normal. This parameter MUST NOT be NULL.

@title: The title of the URI specified by **@url** at the time of the update.

@lastModifiedTime: A timestamp in UTC format specifying the time when the social comment was last modified.

@correlationId: The optional request identifier for the current request.

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

Value	Description
1	Failure: Invalid input or internal error.
0	Successful execution.

Result Sets: MUST NOT return any result sets.

3.1.5.42 **proc_SocialRatings_RefreshAverages**

The **proc_SocialRatings_RefreshAverages** stored procedure is invoked to ensure the protocol server has updated the social rating averages.

```
PROCEDURE proc_SocialRatings_RefreshAverages (  
    @partitionID uniqueidentifier  
    ,@lastRunTime datetime  
    ,@currentRefreshTime datetime = null OUTPUT  
    ,@correlationId uniqueidentifier = null  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@lastRunTime: A timestamp in UTC format specifying the time when the social rating averages were last refreshed.

@currentRefreshTime: A timestamp in UTC format specifying the current time. The protocol server MUST set this value with the current time. The protocol server SHOULD compute social rating averages to this point in time.

@correlationId: The optional request identifier for the current request.

Return Values: An **integer** that MUST be zero.

Result Sets: MUST NOT return any result sets.

3.1.5.43 **proc_SocialTags_GetUsersForTerm**

The **proc_SocialTags_GetUsersForTerm** stored procedure is invoked to get the set of user profiles using a specific term. The term is specified using the **@termID** parameter. The protocol server MUST sort rows in descending order by the last modified date of the corresponding social tags.

```
PROCEDURE proc_SocialTags_GetUsersForTerm (  
    @termID uniqueidentifier  
    ,@viewer_user_recordID bigint = null  
    ,@includePrivate bit = 0  
    ,@partitionID uniqueidentifier  
    ,@url nvarchar(2084) = null  
    ,@top int = 1000  
    ,@correlationId uniqueidentifier = null  
);
```

@termID: The identifier of the term. The value MUST NOT be NULL.

@viewer_user_recordID: The record identifier of a user profile. If the value is not NULL, the result set MUST include private social tags that belong to the specified user profile. If the value is NULL, no additional private comments are included.

@includePrivate: A bit specifying if the result set is to include private social tags. If the value is "1", private social tags are included. If the value is zero ("0"), private social tags are not included, except for those identified by the **@viewer_user_recordID** parameter. The value MUST NOT be NULL.

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@url: The URI. This parameter MUST be a valid URI.

@top: Specifies the maximum size, in rows, requested for the result set. The result set MUST NOT contain more rows than specified by **@top**. This value MUST NOT be NULL or negative.

@correlationId: The optional request identifier for the current request.

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

Value	Description
1	Failure: Invalid input or internal error.
0	Successful execution.

Result Sets:

This stored procedure MUST return a [proc_SocialTags_GetUsersForTerm.ResultSet0](#)

3.1.5.44 proc_SocialTags_MergeTerms

The **proc_SocialTags_MergeTerms** stored procedure is invoked to combine the set of user profiles using a specific term. The term is specified using the **@termID** parameter.

```
PROCEDURE proc_SocialTags_MergeTerms (  
  @partitionID uniqueidentifier  
  ,@mergedTerms xml  
  ,@correlationId uniqueidentifier = null  
  ,@numDeleted int = null OUTPUT  
  ,@numUpdated int = null OUTPUT  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@mergedTerms: A list of similar terms. This value MUST conform to the term **XML schema**.

@correlationId: The optional request identifier for the current request.

@numDeleted: A non-negative integer representing the number of deleted terms MUST be returned.

@numUpdated: A non-negative integer representing the number of updated terms MUST be returned.

Return Values: An **integer** that MUST be zero.

Result Sets: MUST NOT return any result sets.

3.1.5.45 proc_SocialTags_RefreshTagClouds

The **proc_SocialTags_RefreshTagClouds** stored procedure is called to refresh the social tag clouds.

```
PROCEDURE proc_SocialTags_RefreshTagClouds (  

```

```

@partitionID uniqueidentifier
,@lastRunTime datetime
,@currentRefreshTime datetime = null OUTPUT
,@correlationId uniqueidentifier = null
);

```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@lastRunTime: A timestamp in UTC format specifying the time when the social tag clouds were last refreshed. The protocol client MUST set this value to a time prior to the current time. The protocol server SHOULD compute social tag clouds using social tag data more recent than this value.

@currentRefreshTime: A timestamp in UTC format specifying the current time. The protocol server MUST set this value with the current time. The protocol server SHOULD compute social tag clouds to this point in time.

@correlationId: The optional request identifier for the current request.

Return Values: An integer that MUST be zero.

Result Sets: MUST NOT return any result sets.

3.1.5.46 proc_SocialTags_ReplicateForDelete

The **proc_SocialTags_ReplicateForDelete** stored procedure is called to update a social tag with a deletion time.

```

PROCEDURE proc_SocialTags_ReplicateForDelete (
@partitionID uniqueidentifier
,@user_recordID bigint
,@url nvarchar(2084)
,@termID uniqueidentifier
,@deletedTime datetime
,@correlationId uniqueidentifier = null
);

```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@user_recordID: The record identifier of the user profile associated with the social tag. The value MUST NOT be NULL.

@url: The URI associated with the social tag. This parameter MUST be a valid URI.

@termID: The identifier of the term associated with the social tag. The value MUST NOT be NULL.

@deletedTime: The timestamp in UTC format when the social tag was deleted. This value MUST NOT be NULL.

@correlationId: The optional request identifier for the current request.

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

Value	Description
1	Failure: Invalid input or internal error.

Value	Description
0	Successful execution.

Result Sets: MUST NOT return any result sets.

3.1.5.47 **proc_SocialRatings_ReplicateForLive**

The **proc_SocialRatings_ReplicateForLive** stored procedure is called to add or update a social rating that has not been deleted.

```

PROCEDURE proc_SocialRatings_ReplicateForLive (
  @partitionID uniqueidentifier
  ,@user_recordID bigint
  ,@userID uniqueidentifier
  ,@url nvarchar(2084)
  ,@rating int
  ,@lastModifiedTime datetime
  ,@correlationId uniqueidentifier = null
);

```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@user_recordID: The record identifier of the user profile associated with the social rating. The value MUST NOT be NULL.

@userID: The unique identifier of the user profile associated with the social rating. The value MUST NOT be NULL.

@url: The URI associated with the social tag. This parameter MUST be a valid URI.

@rating: An **integer** value indicating the user's social rating of the URI specified by **@url**. This parameter MUST be greater than or equal to zero and less than or equal to 100. This parameter MUST NOT be NULL.

@lastModifiedTime: A timestamp in UTC format specifying the time when the social comment was last modified.

@correlationId: The optional request identifier for the current request.

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

Value	Description
1	Failure: Invalid input or internal error..
0	Successful execution.

Result Sets: MUST NOT return any result sets.

3.1.5.48 **proc_SocialTags_ReplicateForLive**

The **proc_SocialTags_ReplicateForLive** stored procedure is called to add or update a term that has not been deleted.

```

PROCEDURE proc_SocialTags_ReplicateForLive (
  @partitionID uniqueidentifier
  ,@user_recordID bigint
  ,@userID uniqueidentifier
  ,@url nvarchar(2084)
  ,@termID uniqueidentifier
  ,@inputTermLabel nvarchar(255)
  ,@title nvarchar(500) = null
  ,@isPrivate bit
  ,@lastModifiedTime datetime
  ,@correlationId uniqueidentifier = null
);

```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@user_recordID: The record identifier of the user profile associated with the social tag. The value MUST NOT be NULL.

@userID: The unique identifier of the user profile associated with the social tag. The value MUST NOT be NULL.

@url: The URI associated with the social tag. This parameter MUST be a valid URI.

@termID: The identifier of the term associated with the social tag. The value MUST NOT be NULL.

@inputTermLabel: The **string** representation of the input term in the user's **locale** at the time of input.

@title: The title of the URI specified by **@url** at the time of the update. This value MUST NOT be NULL.

@isPrivate: A bit specifying if the social tag is private or public. If the value is "1", the social tag is private. If the value is zero ("0"), the social tag is public. The value MUST NOT be NULL.

@lastModifiedTime: A timestamp in UTC format specifying the time when the social comment was last modified.

@correlationId: The optional request identifier for the current request.

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

Value	Description
1	Failure: Invalid input or internal error.
0	Successful execution.

Result Sets: MUST NOT return any result sets.

3.1.5.49 proc_SocialTags_GetByTime

The **proc_SocialTags_GetByTime** stored procedure is invoked to get the set of terms used by social tags within a specified period of time.

```

PROCEDURE proc_SocialTags_GetByTime (
  @partitionID uniqueidentifier

```



```

, @startTime datetime
, @endTime datetime
, @max int = 1000
, @skip int = 0
, @includePrivate bit = 0
, @correlationId uniqueidentifier = null
);

```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@startTime: A timestamp in UTC format specifying the starting range, exclusive, for the modified date of the comments included in the result set.

@endTime: A timestamp in UTC format specifying the ending range, inclusive, for the modified date of the comments included in the result set.

@max: Specifies the maximum size, in rows, requested for the result set. The result set MUST NOT contain more rows than specified by **@max**. If NULL or unspecified, the protocol server MUST return a maximum of 1000 rows. Otherwise, this parameter MUST be a non-negative integer no greater than 1000.

@skip: Specifies the number of rows matching the criteria specified by the other input parameters that MUST be omitted from the result set.

@includePrivate: A bit specifying if the result set is to include private social tags. If the value is "1", private social tags are included. If the value is zero("0"), private social tags are not included. The value MUST NOT be NULL.

@correlationId: The optional request identifier for the current request.

Return Values: An **integer** that MUST be zero.

Result Sets:

This stored procedure MUST return a [proc_SocialTags_GetByTime.ResultSet0](#)

3.1.5.50 proc_SocialTags_GetChangedUsers

The **proc_SocialTags_GetTerms** stored procedure is invoked to get the set of user record identifiers identifying user profiles that have been changed since **@lastFetchedTime**.

```

PROCEDURE proc_SocialTags_GetChangedUsers (
    @partitionID uniqueidentifier
    , @lastFetchedTime datetime
    , @currentFetchedTime datetime = null OUTPUT
    , @correlationId uniqueidentifier = null
);

```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@lastFetchedTime: The UTC time when changed user profiles were last fetched.

@currentFetchedTime: The current UTC time.

@correlationId: The optional request identifier for the current request.

Return Values: An **integer** that MUST be zero.

Result Sets:

This stored procedure MUST return a [proc_SocialTags_GetChangedUsers.ResultSet0](#)

3.1.5.51 **proc_SocialTags_GetDeletes**

The **proc_SocialTags_GetDeletes** stored procedure is invoked to get the set of terms that were deleted during the specified period of time.

```
PROCEDURE proc_SocialTags_GetDeletes (  
    @partitionID uniqueidentifier  
    ,@startTime datetime  
    ,@endTime datetime  
    ,@correlationId uniqueidentifier = null  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@startTime: A timestamp in UTC format specifying the starting range (inclusive) for the modified date of the comments included in the result set.

@endTime: A timestamp in UTC format specifying the ending range (exclusive) for the modified date of the comments included in the result set.

@correlationId: The optional request identifier for the current request.

Return Values: An integer that MUST be 0.

Result Sets:

This stored procedure MUST return a [proc_SocialTags_GetDeletes.ResultSet0](#)

3.1.5.52 **proc_SocialTags_GetForReplicationByTime**

The **proc_SocialTags_GetForReplicationByTime** stored procedure is invoked to get the set of terms changes made during the specified period of time.

```
PROCEDURE proc_SocialTags_GetForReplicationByTime (  
    @partitionID uniqueidentifier  
    ,@startTime datetime  
    ,@endTime datetime  
    ,@correlationId uniqueidentifier = null  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@startTime: A timestamp in UTC format specifying the starting range, inclusive, for the modified date of the comments included in the result set.

@endTime: A timestamp in UTC format specifying the ending range, exclusive, for the modified date of the comments included in the result set.

@correlationId: The optional request identifier for the current request.

Return Values: An **integer** that MUST be zero.

Result Sets:

This stored procedure MUST return a [proc_SocialTags_GetForReplicationByTime.ResultSet0](#)

3.1.5.53 **proc_SocialTags_GetForReplicationByUser**

The **proc_SocialTags_GetForReplicationByUser** stored procedure is invoked to get the set of term changes made by a specific social profile.

```
PROCEDURE proc_SocialTags_GetForReplicationByUser (  
    @partitionID uniqueidentifier  
    ,@user_RecordID int  
    ,@correlationId uniqueidentifier = null  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@user_RecordID: A record identifier of the user profile associated with the social comment. This parameter MUST NOT be NULL.

@correlationId: The optional request identifier for the current request.

Return Values: An **integer** that MUST be zero.

Result Sets:

This stored procedure MUST return a [proc_SocialTags_GetForReplicationByUser.ResultSet0](#)

3.1.5.54 **proc_SocialTags_GetForUrlByUser**

The **proc_SocialTags_GetForUrlByUser** stored procedure is invoked to get the set of terms used by a specific social profile on a specific URI.

```
PROCEDURE proc_SocialTags_GetForUrlByUser (  
    @partitionID uniqueidentifier  
    ,@user_recordID bigint  
    ,@url nvarchar(2084)  
    ,@includePrivate bit  
    ,@top int = 1000  
    ,@correlationId uniqueidentifier = null  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@user_recordID: The record identifier of the user profile associated with the social tag. The value MUST NOT be NULL.

@url: The URI associated with the social tag. This parameter MUST be a valid URI.

@includePrivate: A bit specifying if the result set is to include private social tags. If the value is "1", private social tags are included. If the value is zero ("0"), private social tags are not included. The value MUST NOT be NULL.

@top: Specifies the maximum size, in rows, requested for the result set. The result set MUST NOT contain more rows than specified by **@top**. This value MUST NOT be NULL, negative, or greater than 3000.

@correlationId: The optional request identifier for the current request.

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

Value	Description
1	Failure: Invalid input or internal error.
0	Successful execution.

Result Sets:

For the following combination of parameters:

- **@partitionID:** 0c37852b-34d0-418e-91c6-2ac25af4be5b
- **@user_recordID:** 102
- **@url:** Select **Url** from **Urls**.

This stored procedure MUST return a [proc_SocialTags_GetTermsForUser.ResultSet0](#)

3.1.5.55 proc_SocialTags_GetForUrlByTime

The **proc_SocialTags_GetForUrlByTime** stored procedure is invoked to get the set of terms associated with a specific URI during a specified period of time.

```
PROCEDURE proc_SocialTags_GetForUrlByTime (  
    @url nvarchar(2084)  
    ,@partitionID uniqueidentifier  
    ,@user_recordID bigint = null  
    ,@fromDate datetime = null  
    ,@toDate datetime = null  
    ,@top int = 100  
    ,@correlationId uniqueidentifier = null  
);
```

@url: The URI associated with the social tag. This parameter MUST be a valid URI.

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@user_recordID: The record identifier of the user profile associated with the social tag. The value MUST NOT be NULL.

@fromDate: A timestamp in UTC format specifying the starting range, inclusive, for the modified date of the comments included in the result set. If this parameter is NULL, it MUST be ignored.

@toDate: A timestamp in UTC format specifying the ending range, exclusive, for the modified date of the comments included in the result set. If this parameter is NULL, it MUST be ignored.

@top: Specifies the maximum size, in rows, requested for the result set. The result set MUST NOT contain more rows than specified by **@top**. This value MUST NOT be NULL, negative, or greater than 100.

@correlationId: The optional request identifier for the current request.

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

Value	Description
1	Failure: Invalid input or internal error.
0	Successful execution.

Result Sets:

This stored procedure MUST return a [proc_SocialTags_GetForTerms.ResultSet0](#)

3.1.5.56 proc_SocialTags_GetTermsForUrlBase

The **proc_SocialTags_GetTermsForUrlBase** stored procedure is invoked to get the set of terms associated with a specific **base URI**.

```
PROCEDURE proc_SocialTags_GetTermsForUrlBase (  
  @partitionID uniqueidentifier  
  ,@url nvarchar(2084)  
  ,@top int = 1000  
  ,@correlationId uniqueidentifier = null  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@url: The URI associated with the base URI. This parameter MUST be a valid URI.

@top: Specifies the maximum size, in rows, requested for the result set. The result set MUST NOT contain more rows than specified by **@top**. This value MUST NOT be NULL, negative, or greater than 3000.

@correlationId: The optional request identifier for the current request.

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

Value	Description
1	Failure: Invalid input or internal error.
0	Successful execution.

Result Sets:

For the following combination of parameters:

- **@partitionID:** 0c37852b-34d0-418e-91c6-2ac25af4be5b
- **@url:** Select **Url** from **Urls**.

This stored procedure MUST return a [proc_SocialTags_GetTerms.ResultSet0](#)

3.1.5.57 **proc_SocialTags_GetUrlCountsForTerm**

The **proc_SocialTags_GetUrlCountsForTerm** stored procedure is invoked to get the number of unique URIs that are associated with the specified term.

```
PROCEDURE proc_SocialTags_GetUrlCountsForTerm (  
  @partitionID uniqueidentifier  
  ,@viewer_user_recordID bigint = null  
  ,@includePrivate bit = 0  
  ,@termID uniqueidentifier  
  ,@from datetime = null  
  ,@correlationId uniqueidentifier = null  
  ,@top int = 1000  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@viewer_user_recordID: The record identifier of a user profile. If the value is not NULL, the result set MUST include private social tags that belong to the specified user profile. If the value is NULL, no additional private comments are included.

@includePrivate: A bit specifying if the result set is to include private social tags. If the value is "1", private social tags are included. If the value is zero ("0"), private social tags are not included, except for those identified by the *@viewer_user_recordID* parameter. The value MUST NOT be NULL.

@termID: The identifier of the terms. This value MUST NOT be NULL.

@from: Date and time in UTC format used to filter the result set. If the value is not NULL, URIs included in the result set MUST be associated with social tags with a last modified time greater than or equal to this value. If the value is NULL, there is no restriction on the last modified time.

@correlationId: The optional request identifier for the current request.

@top: Specifies the maximum size, in rows, requested for the result set. The result set MUST NOT contain more rows than specified by **@top**. This value MUST NOT be NULL or negative.

Return Values: An **integer** that MUST be zero.

Result Sets:

This stored procedure MUST return a [proc_SocialTags_GetUrlCountsForTerm.ResultSet0](#)

3.1.5.58 **proc_SocialRatings_GetByTime**

The **proc_SocialRatings_GetByTime** stored procedure is called to retrieve the social ratings made during a specified period of time.

```
PROCEDURE proc_SocialRatings_GetByTime (  
  @partitionID uniqueidentifier  
  ,@startTime datetime  
  ,@endTime datetime  
  ,@max int = 1000  
  ,@skip int = 0  
  ,@correlationId uniqueidentifier = null  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@startTime: A timestamp in UTC format specifying the starting range, exclusive, for the modified date of the social ratings included in the result set.

@endTime: A timestamp in UTC format specifying the ending range, inclusive, for the modified date of the social ratings included in the result set.

@max: Specifies the maximum size, in rows, requested for the result set. The result set MUST NOT contain more rows than specified by **@max**. This parameter MUST be greater than or equal to zero and MUST be less than or equal to 2000.

@skip: Specifies the number of rows matching the criteria specified by the other input parameters that MUST be omitted from the result set.

@correlationId: The optional request identifier for the current request.

Return Values: An **integer** that MUST be zero.

Result Sets:

This stored procedure MUST return a [proc_SocialRatings_GetByTime.ResultSet0](#)

3.1.5.59 **proc_SocialRatings_GetDeletes**

The **proc_SocialRatings_GetDeletes** stored procedure is called to retrieve the social ratings that were deleted during a specified period of time.

```
PROCEDURE proc_SocialRatings_GetDeletes (  
    @partitionID uniqueidentifier  
    ,@startTime datetime  
    ,@endTime datetime  
    ,@correlationId uniqueidentifier = null  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@startTime: A timestamp in UTC format specifying the starting range, inclusive, for the modified date of the social ratings included in the result set.

@endTime: A timestamp in UTC format specifying the ending range, exclusive, for the modified date of the social ratings included in the result set.

@correlationId: The optional request identifier for the current request.

Return Values: An **integer** that MUST be zero.

Result Sets:

This stored procedure MUST return a [proc_SocialRatings_GetDeletes.ResultSet0](#)

3.1.5.60 **proc_SocialRatings_GetForReplicationByTime**

The **proc_SocialRatings_Get** stored procedure is called to retrieve the social ratings that were last modified during a specified time period.

```

PROCEDURE proc_SocialRatings_GetForReplicationByTime (
  @partitionID uniqueidentifier
  ,@startTime datetime
  ,@endTime datetime
  ,@correlationId uniqueidentifier = null
);

```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@startTime: A timestamp in UTC format specifying the starting range, inclusive, for the modified date of the social ratings included in the result set.

@endTime: A timestamp in UTC format specifying the ending range, exclusive, for the modified date of the social ratings included in the result set.

@correlationId: The optional request identifier for the current request.

Return Values: An **integer** that MUST be zero.

Result Sets:

This stored procedure MUST return a [proc_SocialRatings_GetByTime.ResultSet0](#)

3.1.5.61 proc_SocialRatings_GetForReplicationByUser

The **proc_SocialRatings_GetForReplicationByUser** stored procedure is called to retrieve the social ratings given by a specified user profile.

```

PROCEDURE proc_SocialRatings_GetForReplicationByUser (
  @partitionID uniqueidentifier
  ,@user_RecordID int
  ,@correlationId uniqueidentifier = null
);

```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@user_RecordID: A record identifier of the user profile associated with the social rating. This parameter MUST NOT be NULL.

@correlationId: The optional request identifier for the current request.

Return Values: An **integer** that MUST be zero.

Result Sets:

This stored procedure MUST return a [proc_SocialRatings_GetByTime.ResultSet0](#)

3.1.5.62 proc_SocialRatings_ReplicateForDelete

The **proc_SocialRatings_ReplicateForDelete** stored procedure is called to delete a social rating with a specified deletion time. This stored procedure is expected to be called only when the social rating data is replicated from one protocol server to another protocol server.

```

PROCEDURE proc_SocialRatings_ReplicateForDelete (
  @partitionID uniqueidentifier

```



```
,@user_recordID bigint
,@url nvarchar(2084)
,@deletedTime datetime
,@correlationId uniqueidentifier = null
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@user_recordID: A record identifier of the user profile associated with the social rating. This parameter MUST NOT be NULL.

@url: A **string** specifying the URI for which the social rating has to be deleted. This parameter MUST be a valid URI.

@deletedTime: Specifies a timestamp in UTC format when the social rating was deleted. This value MUST NOT be NULL. The social rating that is to be deleted MUST have a last modified time less than **@deletedTime**.

@correlationId: The optional request identifier for the current request.

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

Value	Description
1	Failure: Invalid input or internal error.
0	Successful execution.

Result Sets: MUST NOT return any result sets.

3.1.5.63 **proc_Admin_RemoveSocialItems**

The **proc_Admin_RemoveSocialItems** stored procedure is called to delete old social tags, social comments, and social ratings.

```
PROCEDURE proc_Admin_RemoveSocialItems (
@EndDate datetime
,@PartitionID uniqueidentifier = null
,@RemoveTags bit = 0
,@RemoveComments bit = 0
,@RemoveRatings bit = 0
,@NumTagsRemoved int = null OUTPUT
,@NumCommentsRemoved int = null OUTPUT
,@NumRatingsRemoved int = null OUTPUT
,@NumUrlsRemoved int = null OUTPUT
,@CorrelationID uniqueidentifier = null
);
```

@EndDate: Specifies a timestamp in UTC format. Specified social data that was last modified prior to **@EndDate** MUST be deleted.

@PartitionID: A GUID used to filter the current request. If specified, the protocol server MUST delete social data only from the specified partition. If NULL, the protocol server MUST delete social data from all partitions.

@RemoveTags: A bit specifying if relevant social tags MUST be deleted. If the value is 1, relevant social tags MUST be deleted. If the value is 0, the protocol server MUST NOT delete social tags. The value MUST NOT be set to NULL.

@RemoveComments: A bit specifying if relevant social comments MUST be deleted. If the value is 1, relevant social comments MUST be deleted. If the value is 0, the protocol server MUST NOT delete social tags. The value MUST NOT be set to NULL.

@RemoveRatings: A bit specifying if relevant social ratings MUST be deleted. If the value is 1, relevant social ratings MUST be deleted. If the value is 0, the protocol server MUST NOT delete social tags. The value MUST NOT be set to NULL.

@NumTagsRemoved: The protocol server MUST return the number of social tags that were deleted.

@NumCommentsRemoved: The protocol server MUST return the number of social comments that were deleted.

@NumRatingsRemoved: The protocol server MUST return the number of social ratings that were deleted.

@NumUrlsRemoved: The protocol server MUST return the number of URIs that are no longer being referenced by social data.

@CorrelationID: The optional request identifier for the current request.

Return Values: An integer that MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.5.64 **proc_Admin_CompactChangeLogs**

The **proc_Admin_CompactChangeLogs** stored procedure is called to perform periodic maintenance on the protocol server's change log.

```
PROCEDURE proc_Admin_CompactChangeLogs (  
    @partitionID uniqueidentifier  
    ,@correlationId uniqueidentifier = null  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@correlationId: The optional request identifier for the current request.

Return Values: An **integer** that MUST be zero.

Result Sets: MUST NOT return any result sets.

3.1.5.65 **proc_SocialData_BulkDeleteForUser**

The **proc_SocialData_BulkDeleteForUser** stored procedure is called to delete all social data for a specific user.

```
PROCEDURE proc_SocialData_BulkDeleteForUser (  
    @partitionID uniqueidentifier  
    ,@user_recordID bigint
```

```
,@correlationId uniqueidentifier = null
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@user_recordID: A record identifier of the user profile associated with the social data. This parameter MUST NOT be NULL.

@correlationId: The optional request identifier for the current request.

Return Values: An **integer** that MUST be zero.

Result Sets: MUST NOT return any result sets.

3.1.5.66 **proc_Social_SetSecurityTrimmerEnabled**

The **proc_Social_SetSecurityTrimmerEnabled** stored procedure is called to update a setting indicating whether security trimming heuristics are enabled.

```
PROCEDURE proc_Social_SetSecurityTrimmerEnabled (
    @partitionID uniqueidentifier
    ,@securityTrimmerEnabled bit
    ,@correlationId uniqueidentifier = null
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@securityTrimmerEnabled: A bit specifying whether the security trimming heuristics are enabled. If the value is "1", the security trimming heuristics MUST be enabled for all social data. If the value is zero ("0"), the security trimming heuristics MUST be disabled for all social data. If the value is NULL, the security trimming heuristics MUST be applied for all social data whose **Uri** values match the result set returned by the **proc_Social_GetSecurityTrimUrls** stored procedure.

@correlationId: The optional request identifier for the current request.

Return Values: An **integer** that MUST be zero.

Result Sets: MUST NOT return any result sets.

3.1.5.67 **proc_Social_RemoveSecurityTrimUrl**

The **proc_Social_RemoveSecurityTrimUrl** stored procedure is called to remove a URI from the set of URIs associated with the security trimming heuristics.

```
PROCEDURE proc_Social_RemoveSecurityTrimUrl (
    @partitionID uniqueidentifier
    ,@url nvarchar(2084)
    ,@alwaysAllow bit
    ,@correlationId uniqueidentifier = null
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@url: The URI to be removed from the set of URIs associated with the security trimming heuristics.

@alwaysAllow: A bit specifying whether the URI is to be removed from the always allowed set. If the value is "1", the URI MUST be removed from the always allowed set. If the value is zero ("0"), the URI MUST be removed from the always security trim set. The value MUST NOT be NULL.

@correlationId: The optional request identifier for the current request.

Return Values: An **integer** that MUST be zero.

Result Sets: MUST NOT return any result sets.

3.1.5.68 **proc_Social_GetSecurityTrimUrls**

The **proc_Social_GetSecurityTrimUrls** stored procedure is called to retrieve the set of URIs associated with the security trimming heuristics.

```
PROCEDURE proc_Social_GetSecurityTrimUrls (  
    @partitionID uniqueidentifier  
    ,@correlationId uniqueidentifier = null  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@correlationId: The optional request identifier for the current request.

Return Values: An **integer** that MUST be zero.

Result Sets:

This stored procedure MUST return a [proc_Social_GetSecurityTrimUrls.ResultSet0](#)

3.1.5.69 **proc_Social_GetSecurityTrimmerEnabled**

The **proc_Social_GetSecurityTrimmerEnabled** stored procedure is called to retrieve a setting indicating whether security trimming heuristics are enabled.

```
PROCEDURE proc_Social_GetSecurityTrimmerEnabled (  
    @partitionID uniqueidentifier  
    ,@securityTrimmerEnabled bit OUTPUT  
    ,@correlationId uniqueidentifier = null  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@securityTrimmerEnabled: An output bit that receives a value specifying whether the security trimming heuristics are enabled. If the value is "1", the security trimming heuristics MUST be enabled for all social data. If the value is zero ("0"), the security trimming heuristics MUST be disabled for all social data. If the value is NULL, the security trimming heuristics MUST be applied for all social data whose **Url** values match the result set returned by the **proc_Social_GetSecurityTrimUrls** stored procedure. The input value MUST be ignored.

@correlationId: The optional request identifier for the current request.

Return Values: An **integer** that MUST be zero.

Result Sets: MUST NOT return any result sets.

3.1.5.70 proc_Social_AddSecurityTrimUrl

The **proc_Social_AddSecurityTrimUrl** stored procedure is called to add a URI to the set of URIs associated with the security trimming heuristics.

```
PROCEDURE proc_Social_AddSecurityTrimUrl (  
    @partitionID uniqueidentifier  
    ,@url nvarchar(2084)  
    ,@alwaysAllow bit  
    ,@correlationId uniqueidentifier = null  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@url: The URI to be added to the set of URIs associated with the security trimming heuristics.

@alwaysAllow: A bit specifying whether the URI is to be added to the always allowed set. If the value is "1", the URI MUST be added to the always allowed set. If the value is zero ("0"), the URI MUST be added to the always security trim set. The value MUST NOT be NULL.

@correlationId: The optional request identifier for the current request.

Return Values: An **integer** that MUST be zero.

Result Sets: MUST NOT return any result sets.

3.1.5.71 proc_SocialData_MergeSocialNotes

The **proc_SocialData_MergeSocialNotes** stored procedure is called to move the social notes from one URI to another one. The notes from the old URI will effectively be merged with the notes already present in the new URI.

```
PROCEDURE proc_SocialData_MergeSocialNotes (  
    @partitionID uniqueidentifier  
    ,@oldUrl varchar(2048)  
    ,@newUrl varchar(2048)  
    ,@success bit = null OUTPUT  
    ,@correlationId uniqueidentifier = null  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@oldUrl: The source URI from which the social notes will be moved.

@newUrl: The target URI to which the social notes will be moved.

@success: Output parameter specifying if the operation completed successfully or failed:

Value	Description
0	Failure.
1	Success.

@correlationId: The optional request identifier for the current request.

Return Values: An **integer** that MUST be zero.

Value	Description
-------	-------------

Result Sets: MUST NOT return any result sets.

3.1.5.72 **proc_SocialData_MigrateSocialDataUri**

The **proc_SocialData_MigrateSocialDataUri** stored procedure is called to migrate the social data from one URL to another one.

```
PROCEDURE proc_SocialData_MigrateSocialDataUri (  
    @partitionID uniqueidentifier  
    ,@oldUri varchar(2048)  
    ,@newUri varchar(2048)  
    ,@success bit = null OUTPUT  
    ,@correlationId uniqueidentifier = null  
);
```

@partitionID: A GUID used to filter the current request. This value MUST NOT be NULL or empty.

@oldUri: The source URI from which the social data will be moved.

@newUri: The target URI to which the social data will be moved.

@success: Output parameter specifying if the operation completed successfully or failed:

Value	Description
0	Failure.
1	Success.

@correlationId: The optional request identifier for the current request.

Return Values: An **integer** that MUST be zero.

Value	Description
-------	-------------

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.

3.2.1 Abstract Data Model

None.

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Higher-Layer Triggered Events

None.

3.2.5 Message Processing Events and Sequencing Rules

None.

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

4 Protocol Examples

The following examples show typical interactions between the protocol client and the protocol server.

4.1 Add a Social Tag to a URI

In this example, the **current user** visits a page and wants to add a social tag to it. The interface that the protocol client provides for this also displays the past three social tags from the current user and a list of the social tags on the URI from all users.

The protocol client starts out by querying the last 3 social tags from the current user. Consider the following **Transact-Structured Query Language (T-SQL)** request:

```
exec dbo.proc_SocialTags_GetForUser
@partitionID='0C37852B-34D0-418E-91C6-2AC25AF4BE5B',
@correlationId='BC0BE25A-9FB9-4A03-A8A1-20D4B781A675',
@user_recordID=2,
@includePrivate=0,
@top=3
```

The protocol server responds with a return code of 0, indicating successful execution, and the following result set showing the last 3 social tags from the current user. Note that the user only has 1 social tag so only 1 is returned.

```
TermID
D3970499-9FA6-40C7-B592-4DE5B9783030

LastModifiedTime
2009-05-13 20:41:46.317

Title
Team Site - Home

IsPrivate
0

IsExternal
0

Url
http://server.example.com//SitePages
```

When displaying terms to the user, the protocol client uses the associated metadata. For more information, see [\[MS-EMMSTORE\]](#).

The protocol client then asks for the three most recent social tags on the URI. Consider the following T-SQL request for this information:

```
exec dbo.proc_SocialTags_GetForUrl
@viewer_user_recordID=2,
@url=N'http://server.example.com//SitePages',
@partitionID='0C37852B-34D0-418E-91C6-2AC25AF4BE5B',
@correlationId='1A9E2B4D-0C29-43EA-B3C9-D068EE3F7D0A',
```


@top=3

The protocol server responds with a return code of 0, indicating successful execution, and the following result set. Again, note that there are not three social tags to return, so only the available ones are returned.

```
TermID
D3970499-9FA6-40C7-B592-4DE5B9783030
D11DD833-D18A-49EA-8C18-95214CCF182F
User_RecordId
2
1

LastModifiedTime
2009-05-13 20:41:46.317
2009-05-13 20:40:16.563

Title
Team Site - Home
Team Site - Home

IsPrivate
0
0

Url
http://server.example.com//SitePages
http://server.example.com//SitePages
```

The current user now wants to add a social tag to this URI. The protocol client uses the associated metadata to convert the user data to a **termID**. For more information, see [MS-EMMSTORE]. Consider the following T-SQL syntax that adds a social tag:

```
declare @p10 datetime
set @p10='2009-05-13 20:47:42.0670000'
exec dbo.proc_SocialTags_Add
@partitionID='0C37852B-34D0-418E-91C6-2AC25AF4BE5B',
@correlationId='74D7A503-824F-44A7-BCA5-2C582BF623EB',
@user_recordID=2,
@userID='CA448986-2185-4BEF-9331-47A095B8F281',
@url=N'http://co745-25//SitePages',
@termID='306EF734-16B4-4B9B-AA88-6AA1D863C0D2',
@isPrivate=0,
@isExternal=0,
@title=N'Team Site - Home',
@lastModifiedTime=@p10 output
select @p10
```

The protocol server stores this information, responds with a return code of 0, indicating successful execution and gives the following output:

```
LastModifiedTime
2009-05-13 21:33:24.243
```

4.2 Add a Social Comment to a URI

In this example, the current user wants to add a social comment to a URI. The user interface provided by the protocol client in the example will also show the user the three most recent social comments that they have made and the three most recent social comments on the URI, as well as the total number of social comments on the URI.

Please consider the following T-SQL syntax for displaying the three most recent social comments from the current user:

```
exec dbo.proc_SocialComments_GetForUser
@partitionID='0C37852B-34D0-418E-91C6-2AC25AF4BE5B',
@correlationId='BB1511F8-6258-4C12-87DA-FC67B8F84C53',
@user_recordID=2,
@max=3
```

The protocol server returns a code of 0, indicating successful execution, and the following result set:

```
CommentID
2
1

Url
http://server.example.com/my/Person.aspx?accountname=DOMAIN%5CUSER1
http://server.example.com//SitePages

LastModifiedTime
2009-05-13 21:54:15.333
2009-05-13 21:52:56.300

Comment
My profile is very useful for displaying information about myself to other users who can use
it to collaborate with me on parallel professional interests.
I found this site very useful.

IsHighPriority
0
0

Title
DOMAIN\USER1
Team Site - Home

IsExternal
0
0
```

The protocol client then requests the three most recent social comments on the current URI. Consider the following T-SQL syntax:

```
exec dbo.proc_SocialComments_GetForUrl
@partitionID='0C37852B-34D0-418E-91C6-2AC25AF4BE5B',
@correlationId='64859F69-5856-41B7-98CE-2F20D45412C0',
@url=N'http://server.example.com//SitePages',
@max=3
```

The protocol server responds with a return code of 0, indicating successful execution, and the following result set:

```
CommentID
1

User_RecordID
2

LastModifiedTime
2009-05-13 21:52:56.300

Comment
I found this site very useful.

IsHighPriority
0

Title
Team Site - Home
```

The protocol client then requests the total number of social comments on the URI. Consider the following T-SQL syntax:

```
declare @p4 int
set @p4=2
exec dbo.proc_SocialComments_Count
@partitionID='0C37852B-34D0-418E-91C6-2AC25AF4BE5B',
@correlationId='D17B1E99-A695-410E-A1A1-7C372427FA4F',
@url=N'http://co745-25//SitePages',
@count=@p4 output
select @p4
```

The protocol client returns a code of 0, indicating successful execution, and the following as output:

```
count
1
```

Now the current user adds their social comment. The protocol client uses the following T-SQL:

```
declare @p10 bigint
set @p10=3
declare @p11 datetime
set @p11='2009-05-13 22:03:58.2600000'
exec dbo.proc_SocialComments_Add @partitionID='0C37852B-34D0-418E-91C6-2AC25AF4BE5B',
@correlationId='D17B1E99-A695-410E-A1A1-7C372427FA4F',
@user_recordID=2,
@userID='CA448986-2185-4BEF-9331-47A095B8F281',
@url=N'http://server.example.com//SitePages',
@comment=N'This site contains some important information.',
@isHighPriority=0,@title=N'Team Site - Home',
@isExternal=0,
@commentID=@p10 output,
@lastModifiedTime=@p11 output
```

```
select @p10, @p11
```

The protocol server stores this information and returns a code of 0, indicating successful execution. It also returns the following values as output:

```
commentID
4

lastModifiedTime
2009-05-13 22:28:23.817
```

4.3 Delete a Social Tag from a URI

In this example the current user deletes one of their social tags. Consider the following T-SQL syntax:

```
exec dbo.proc_SocialTags_Delete
@partitionID='0C37852B-34D0-418E-91C6-2AC25AF4BE5B',
@correlationId='F85EBF2A-5DF0-4D66-B09A-A905400B407C',
@user_recordID=2,
@url=N'http://server.example.com//SitePages',
@termID='306EF734-16B4-4B9B-AA88-6AA1D863C0D2'
```

The protocol server deletes the tag and returns a code of 0 indicating successful execution.

4.4 Mark the Social Tags on a URI as Private

In this example the current user marks their social tags on a URI as private. Consider the following T-SQL syntax:

```
declare @p8 datetime
set @p8=NULL
exec dbo.proc_SocialTags_Update
@partitionID='0C37852B-34D0-418E-91C6-2AC25AF4BE5B',
@correlationId='8F524904-CD8D-4C00-8C03-44B7DE88D06A',
@user_recordID=2,
@url=N'http://server.example.com//SitePages',
@termID='D3970499-9FA6-40C7-B592-4DE5B9783030',
@isPrivate=1,
@lastModifiedTime=@p8 output
select @p8
```

The protocol server performs the requested update and returns a code of 0, indicating successful execution. As output it returns **lastModifiedTime** as NULL (because **needUpdateTime** is 0).

4.5 Delete a Social Comment from a URI

In this example, the current user deletes a particular social comment for which the protocol client has already stored the identifier. Consider the following T-SQL syntax:

```
exec dbo.proc_SocialComments_Delete
@partitionID='0C37852B-34D0-418E-91C6-2AC25AF4BE5B',
```

```
@correlationId='C78282AB-CFC6-4C8B-B73A-A53E37400733',  
@commentID=4
```

The protocol server returns a code of 0.

4.6 Add a Social Rating to a URI

In this example the current user, who has not added any social ratings, views a page that displays the average social rating of a URI and the current user's social rating of that URI. The current user then adds a social rating for that URI.

Consider the following T-SQL the protocol client sends to get the average social rating.

```
declare @p4 float  
set @p4=90  
declare @p5 datetime  
set @p5='2009-05-13 23:55:03.9200000'  
declare @p6 bigint  
set @p6=2  
exec dbo.proc_SocialRatings_GetAverage  
@partitionID='0C37852B-34D0-418E-91C6-2AC25AF4BE5B',  
@correlationId='A65B652D-A8AC-4C45-A3FA-D94D52FB7F92',  
@url=N'http://server.example.com/Shared%20Documents/Script2.docx',  
@ratingAvg=@p4 output,  
@lastModifiedTime=@p5 output,  
@ratingCount=@p6 output  
select @p4, @p5, @p6
```

The protocol server returns a code of 0, indicating successful execution, and then provides the following values as output:

```
ratingAvg  
66.6666666666667  
  
lastModifiedTime  
2009-05-12 00:02:54.443  
  
ratingCount  
3
```

The protocol client then requests the social rating that the current user has provided for the given URI. Consider the following T-SQL:

```
declare @p5 int  
set @p5=NULL  
declare @p6 datetime  
set @p6=NULL  
exec dbo.proc_SocialRatings_Get  
@partitionID='0C37852B-34D0-418E-91C6-2AC25AF4BE5B',  
@correlationId='A65B652D-A8AC-4C45-A3FA-D94D52FB7F92',  
@user_recordID=3, @url=N'http://server.example.com/Shared%20Documents/Script2.docx',  
@rating=@p5 output,  
@lastModifiedTime=@p6 output
```

```
select @p5, @p6
```

The protocol server returns the following as output, given that the current user has not yet added a social rating to the URI. It also returns 0, indicating successful execution.

```
rating
NULL

lastModifiedTime
NULL
```

The current user now adds a social rating to the URI. Consider the following T-SQL that is sent by the protocol client:

```
declare @p7 datetime
set @p7='2009-05-14 00:02:54.4430000'
exec dbo.proc_SocialRatings_Add
@partitionID='0C37852B-34D0-418E-91C6-2AC25AF4BE5B',
@correlationId='2BD1EFC5-0975-40E1-96C0-0794B1BD1DFE',
@user_recordID=3,
@userID='620DE88A-8012-45B0-B70F-B7A80A8ACC6C',
@url=N'http://server.example.com/Shared%20Documents/Script2.docx',
@rating=20,
@lastModifiedTime=@p7 output
select @p7
```

The protocol server stores this information and returns 0, indicating successful execution. It also returns the following as output:

```
lastModifiedTime
2009-05-14 00:20:38.220
```

4.7 Replicate Social Data

In this example, the protocol client replicates social data from one protocol server to another protocol server. First, the protocol client collects data to be replicated. Consider the following T-SQL call to **proc_SocialTags_GetForReplicationByTime**.

```
exec dbo.proc_SocialTags_GetForReplicationByTime
@partitionID='0C37852B-34D0-418E-91C6-2AC25AF4BE5B',
@correlationId='BC0BE25A-9FB9-4A03-A8A1-20D4B781A675',
@startTime='2009-05-13 00:00:00',
@endTime=' 2009-05-14 24:59:59'
```

The protocol server responds with the following data:

```
User_RecordId
2
1
Url
http://server.example.com//SitePages
http://server.example.com//SitePages
```

```

TermID
D3970499-9FA6-40C7-B592-4DE5B9783030
D11DD833-D18A-49EA-8C18-95214CCF182F
InputTermLabel
Tag
Tag2
Title
Team Site - Home
Team Site - Home

IsPrivate
0
0

LastModifiedTime
2009-05-13 20:41:46.317
2009-05-13 20:40:16.563

```

Now that the protocol client has replication data for social tags, it needs replication data for social comments. Consider the following T-SQL call that is made by the protocol client for **proc_SocialComments_GetForReplicationByTime**.

```

exec dbo.proc_SocialComments_GetForReplicationByTime
@partitionID='0C37852B-34D0-418E-91C6-2AC25AF4BE5B',
@correlationId='BC0BE25A-9FB9-4A03-A8A1-20D4B781A675',
@startTime='2009-05-13 00:00:00',
@endTime=' 2009-05-14 24:59:59'

```

The protocol server responds with the following result set.

```

CommentID
2
1
User_RecordID
1
2
Url
http://server.example.com/my/Person.aspx?accountname=DOMAIN%5CUSER1
http://server.example.com//SitePages

LastModifiedTime
2009-05-13 21:54:15.333
2009-05-13 21:52:56.300

Comment
My profile is very useful for displaying information about myself to other users who can use
it to collaborate with me on parallel professional interests.
I found this site very useful.

IsHighPriority
0
0

Title
DOMAIN\USER1

```

The protocol client now makes the following T-SQL call to `proc_SocialRatings_GetForReplicationByTime` to get replication data for social ratings.

```
exec dbo.proc_SocialRatings_GetForReplicationByTime
@partitionID='0C37852B-34D0-418E-91C6-2AC25AF4BE5B',
@correlationId='BC0BE25A-9FB9-4A03-A8A1-20D4B781A675',
@startTime='2009-05-13 00:00:00',
@endTime=' 2009-05-14 24:59:59'
```

The protocol server responds with the following result set.

```
User_RecordID
1
Url
http://server.example.com/Shared%20Documents/Script2.docx

LastModifiedTime
2009-05-14 00:20:38.220

Rating
20

Title
Script
```

The protocol client now uses metadata and stored procedures to retrieve necessary user data and term data from the original protocol server so that equivalent terms and users can be utilized on the destination protocol server. For more information, see [\[MS-EMMSTORE\]](#) and [\[MS-UPSPROF\]](#). Then on the destination protocol server, the protocol client calls **`proc_SocialTags_ReplicateForLive`**. Consider the following T-SQL syntax for this call.

```
exec dbo.proc_SocialTags_ReplicateForLive
@partitionID='0C37852B-34D0-418E-91C6-2AC25AF4BE5B',
@user_recordID=2,
@userID='CA448986-2185-4BEF-9331-47A095B8F281',
@url=N'http://server.example.com//SitePages',
@termID='306EF734-16B4-4B9B-AA88-6AA1D863C0D2',
@inputTermLabel=Tag,
@title='Team Site - Home',
@isPrivate=0,
@lastModifiedTime='2009-05-13 20:41:46.317',
@correlationId='74D7A503-824F-44A7-BCA5-2C582BF623EB'
```

The destination protocol server stores this information and gives a return code of 1, which is ignored by the protocol client. The protocol client repeats this for all available social tags in this case one more. Consider the following T-SQL syntax for this call.

```
exec dbo.proc_SocialTags_ReplicateForLive
@partitionID='0C37852B-34D0-418E-91C6-2AC25AF4BE5B',
@user_recordID=1,
@userID='620DE88A-8012-45B0-B70F-B7A80A8ACC6C ',
```



```
@url=N'http://server.example.com//SitePages',
@termID= D11DD833-D18A-49EA-8C18-95214CCF182F',
@inputTermLabel=Tag2,
@title='Team Site - Home',
@isPrivate=0,
@lastModifiedTime='2009-05-13 20:40:16.563',
@correlationId='74D7A503-824F-44A7-BCA5-2C582BF623EB'
```

The protocol server stores this data and gives a return code of 1, which is ignored. The protocol client has now replicated available social tags, and can now move on to social comments. Consider the following T-SQL syntax describing the call to **proc_SocialComments_ReplicateForLive**.

```
exec dbo.proc_SocialComments_ReplicateForLive
@partitionID='0C37852B-34D0-418E-91C6-2AC25AF4BE5B',
@user_recordID=1,
@userID='0C37852B-34D0-418E-91C6-2AC25AF4BE5B',
@url=N'http://server.example.com/my/Person.aspx?accountname=DOMAIN%5CUSER1',
@comment='My profile is very useful for displaying information about myself to other users
who can use it to collaborate with me on parallel professional interests.',
@isHighPriority=0,
@title='DOMAIN\USER1',
@lastModifiedTime='2009-05-13 21:54:15.333',
@correlationId='BC0BE25A-9FB9-4A03-A8A1-20D4B781A675'
```

The protocol server stores this data and gives a return code of 1, which is ignored. The protocol client proceeds now to replicate the next social comment. Consider the following T-SQL syntax for this.

```
exec dbo.proc_SocialComments_ReplicateForLive
@partitionID='0C37852B-34D0-418E-91C6-2AC25AF4BE5B',
@user_recordID=2,
@userID='CA448986-2185-4BEF-9331-47A095B8F281',
@url=N'http://server.example.com//SitePages',
@comment='I found this site very useful.',
@isHighPriority=0,
@title='Team Site - Home',
@lastModifiedTime='2009-05-13 21:52:56.300',
@correlationId='BC0BE25A-9FB9-4A03-A8A1-20D4B781A675'
```

The protocol server gives a return code of 1, which is ignored by the protocol client. The protocol client now moves on to replicating social ratings. Consider the following T-SQL call to **proc_SocialRatings_ReplicateForLive**.

```
exec dbo.proc_SocialRatings_ReplicateForLive
@partitionID='0C37852B-34D0-418E-91C6-2AC25AF4BE5B',
@user_recordID=1,
@userID='0C37852B-34D0-418E-91C6-2AC25AF4BE5B',
@url=N'http://server.example.com/Shared%20Documents/Script2.docx',
@rating=20,
@lastModifiedTime='2009-05-14 00:20:38.220',
@correlationId='BC0BE25A-9FB9-4A03-A8A1-20D4B781A675'
```

The protocol server stores this data and gives a return code of 1, which is ignored. To finalize the replication of social ratings the protocol client makes the following T-SQL call.

```
exec dbo.proc_SocialRatings_RefreshAverages  
@partitionID='0C37852B-34D0-418E-91C6-2AC25AF4BE5B',  
@lastRunTime='2009-05-14 00:20:38.220',  
@correlationId='BC0BE25A-9FB9-4A03-A8A1-20D4B781A675'
```

The protocol server returns a value of 0, which is ignored. The protocol client has now replicated all available social data and has finished the process.

5 Security

5.1 Security Considerations for Implementers

None.

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.

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

A

Abstract data model
 [client](#) 71
 [server](#) 22
[Add a social comment to a URI example](#) 74
[Add a social rating to a URI example](#) 77
[Add a social tag to a URI example](#) 72
[Applicability](#) 9
[Attribute groups - overview](#) 21
[Attributes - overview](#) 21

B

[Binary structures - overview](#) 10
[Bit fields - overview](#) 10

C

[Capability negotiation](#) 9
[Change tracking](#) 85
Client
 [abstract data model](#) 71
 [details](#) 70
 [higher-layer triggered events](#) 71
 [initialization](#) 71
 [local events](#) 71
 [message processing](#) 71
 [sequencing rules](#) 71
 [timer events](#) 71
 [timers](#) 71
Common data types
 [overview](#) 10
[Complex types - overview](#) 20

D

Data model - abstract
 [client](#) 71
 [server](#) 22
Data types
 [common](#) 10
Data types - simple
 [overview](#) 10
[Delete a social comment from a URI example](#) 76
[Delete a social tag from a URI example](#) 76
Details
 [social comments \(notes\)](#) 22
 [social ratings](#) 22
 [social tags](#) 22

E

[Elements - overview](#) 20
Events
 [local - client](#) 71
 [local - server](#) 70
 [timer - client](#) 71
 [timer - server](#) 70

Example
 [delete a social comment from a URI](#) 76
 [mark the social tags on a URI as private](#) 76
Examples
 [add a social comment to a URI](#) 74
 [add a social rating to a URI](#) 77
 [add a social tag to a URI](#) 72
 [delete a social tag from a URI](#) 76
 [overview](#) 72
 [replicate social data](#) 78

F

[Fields - vendor-extensible](#) 9
[Flag structures - overview](#) 10

G

[Glossary](#) 7
[Groups - overview](#) 21

H

Higher-layer triggered events
 [client](#) 71
 [server](#) 22

I

[Implementer - security considerations](#) 83
[Index of security parameters](#) 83
[Informative references](#) 8
Initialization
 [client](#) 71
 [server](#) 22
[Introduction](#) 7

L

Local events
 [client](#) 71
 [server](#) 70

M

[Mark the social tags on a URI as private example](#) 76
Message processing
 [client](#) 71
 [server](#) 23
Messages
 [attribute groups](#) 21
 [attributes](#) 21
 [binary structures](#) 10
 [bit fields](#) 10
 [common data types](#) 10
 [complex types](#) 20
 [elements](#) 20
 [enumerations](#) 10
 [flag structures](#) 10

groups	21	proc Admin CompactChangeLogs	66
namespaces	20	proc Admin DeletePartition	23
proc Social GetUsers.ResultSet0 result set	14	proc Admin RemoveSocialItems	65
proc SocialComments GetForUrl.proc SocialComments GetForUrl.Default.ResultSet0 result set	14	proc Admin SetupPartition	23
proc SocialComments GetForUrlBase.ResultSet0 result set	10	proc Social AddSecurityTrimUrl	69
proc SocialComments GetForUser.ResultSet0 result set	11	proc Social GetSecurityTrimmerEnabled	68
proc SocialComments GetForUserWithTag.ResultSet0 result set	11	proc Social GetSecurityTrimUrls	68
proc SocialComments GetDeletes.ResultSet0 result set	16	proc Social GetUsers	47
proc SocialRatings GetByTime.ResultSet0 result set	19	proc Social RemoveSecurityTrimUrl	67
proc SocialRatings GetDeletes.ResultSet0 result set	20	proc Social SetSecurityTrimmerEnabled	67
proc SocialRatings GetForUrl.proc SocialRatings GetForUrl.Default.ResultSet0 result set	15	proc SocialComments Add	23
proc SocialRatings GetForUser.ResultSet0 result set	12	proc SocialComments Count	25
proc SocialRatings GetRatingsForUpdate.ResultSet0 result set	12	proc SocialComments Delete	25
proc SocialTags GetByTime.ResultSet0 result set	17	proc SocialComments Get	26
proc SocialTags GetChangedUsers.ResultSet0 result set	17	proc SocialComments GetByTime	47
proc SocialTags GetDeletes.ResultSet0 result set	17	proc SocialComments GetDeletes	48
proc SocialTags GetForReplicationByTime.ResultSet0 result set	18	proc SocialComments GetForReplicationByTime	48
proc SocialTags GetForReplicationByUser.ResultSet0 result set	18	proc SocialComments GetForReplicationByUser	49
proc SocialTags GetForTerms.ResultSet0 result set	13	proc SocialComments GetForUrl	27
proc SocialTags GetForUser.ResultSet0 result set	13	proc SocialComments GetForUrlByTime	49
proc SocialTags GetTerms.ResultSet0 result set	14	proc SocialComments GetUser	28
proc SocialTags GetTermsForUrl.proc SocialTags GetTermsForUrl.Default.ResultSet0 result set	15	proc SocialComments GetUserWithTag	29
proc SocialTags GetTermsForUser.proc SocialTags GetTermsForUser.Default.ResultSet0 result set	16	proc SocialComments ReplicateForDelete	50
proc SocialTags GetTermsForUser.ResultSet0 result set	14	proc SocialComments ReplicateForLive	51
proc SocialTags GetUrlCountsForTerm.ResultSet0 result set	19	proc SocialComments Update	29
proc SocialTags GetUrlsForTerm.ResultSet0 result set	14	proc SocialData BulkDeleteForUser	66
proc SocialTags GetUsersForTerm.ResultSet0 result set	16	proc SocialData MergeSocialNotes	69
simple data types	10	proc SocialData MigrateSocialDataUrl	70
simple types	20	proc SocialRatings Add	30
table structures	20	proc SocialRatings Count	31
transport	10	proc SocialRatings Delete	32
view structures	20	proc SocialRatings DeleteForUser	32
XML structures	20	proc SocialRatings Get	33
Methods		proc SocialRatings GetAverage	33
		proc SocialRatings GetByTime	62
		proc SocialRatings GetDeletes	63
		proc SocialRatings GetForReplicationByTime	63
		proc SocialRatings GetForReplicationByUser	64
		proc SocialRatings GetForUrl	35
		proc SocialRatings GetUser	35
		proc SocialRatings GetLastAverageSyncTime	36
		proc SocialRatings GetRatingsForUpdate	36
		proc SocialRatings RefreshAverages	52
		proc SocialRatings ReplicateForDelete	64
		proc SocialRatings ReplicateForLive	55
		proc SocialRatings SetLastAverageSyncTime	37
		proc SocialTags Add	37
		proc SocialTags Count	38
		proc SocialTags Delete	39
		proc SocialTags DeleteForUser	40
		proc SocialTags GetByTime	56
		proc SocialTags GetChangedUsers	57
		proc SocialTags GetDeletes	58
		proc SocialTags GetForReplicationByTime	58
		proc SocialTags GetForReplicationByUser	59
		proc SocialTags GetForTerms	40
		proc SocialTags GetForUrl	41
		proc SocialTags GetForUrlByTime	60
		proc SocialTags GetForUrlByUser	59

- [proc SocialTags GetForUser](#) 42
- [proc SocialTags GetTerms](#) 43
- [proc SocialTags GetTermsForUrl](#) 43
- [proc SocialTags GetTermsForUrlBase](#) 61
- [proc SocialTags GetTermsForUser](#) 44
- [proc SocialTags GetUrlCountsForTerm](#) 62
- [proc SocialTags GetUrlsForTerm](#) 45
- [proc SocialTags GetUsersForTerm](#) 52
- [proc SocialTags MergeTerms](#) 53
- [proc SocialTags RefreshTagClouds](#) 53
- [proc SocialTags ReplicateForDelete](#) 54
- [proc SocialTags ReplicateForLive](#) 55
- [proc SocialTags Update](#) 46

N

- [Namespaces](#) 20
- [Normative references](#) 8

O

- [Overview \(synopsis\)](#) 8

P

- [Parameters - security index](#) 83
- [Preconditions](#) 9
- [Prerequisites](#) 9
- [proc Admin CompactChangeLogs method](#) 66
- [proc Admin DeletePartition method](#) 23
- [proc Admin RemoveSocialItems method](#) 65
- [proc Admin SetupPartition method](#) 23
- [proc Social AddSecurityTrimUrl method](#) 69
- [proc Social GetSecurityTrimmerEnabled method](#) 68
- [proc Social GetSecurityTrimUrls method](#) 68
- [proc Social GetUsers method](#) 47
- [proc Social GetUsers.ResultSet0 result set](#) 14
- [proc Social RemoveSecurityTrimUrl method](#) 67
- [proc Social SetSecurityTrimmerEnabled method](#) 67
- [proc SocialComments Add method](#) 23
- [proc SocialComments Count method](#) 25
- [proc SocialComments Delete method](#) 25
- [proc SocialComments Get method](#) 26
- [proc SocialComments GetByTime method](#) 47
- [proc SocialComments GetDeletes method](#) 48
- [proc SocialComments GetForReplicationByTime method](#) 48
- [proc SocialComments GetForReplicationByUser method](#) 49
- [proc SocialComments GetForUrl method](#) 27
- [proc SocialComments GetForUrl.proc SocialComments GetForUrl.Default.ResultSet0 result set](#) 14
- [proc SocialComments GetForUrlBase.ResultSet0 result set](#) 10
- [proc SocialComments GetForUrlByTime method](#) 49
- [proc SocialComments GetForUser method](#) 28
- [proc SocialComments GetForUser.ResultSet0 result set](#) 11
- [proc SocialComments GetForUserWithTag method](#) 29
- [proc SocialComments GetForUserWithTag.ResultSet0 result set](#) 11

- [proc SocialComments ReplicateForDelete method](#) 50
- [proc SocialComments ReplicateForLive method](#) 51
- [proc SocialComments Update method](#) 29
- [proc SocialComments GetDeletes.ResultSet0 result set](#) 16
- [proc SocialData BulkDeleteForUser method](#) 66
- [proc SocialData MergeSocialNotes method](#) 69
- [proc SocialData MigrateSocialDataUrl method](#) 70
- [proc SocialRatings Add method](#) 30
- [proc SocialRatings Count method](#) 31
- [proc SocialRatings Delete method](#) 32
- [proc SocialRatings DeleteForUser method](#) 32
- [proc SocialRatings Get method](#) 33
- [proc SocialRatings GetAverage method](#) 33
- [proc SocialRatings GetByTime method](#) 62
- [proc SocialRatings GetByTime.ResultSet0 result set](#) 19
- [proc SocialRatings GetDeletes method](#) 63
- [proc SocialRatings GetDeletes.ResultSet0 result set](#) 20
- [proc SocialRatings GetForReplicationByTime method](#) 63
- [proc SocialRatings GetForReplicationByUser method](#) 64
- [proc SocialRatings GetForUrl method](#) 35
- [proc SocialRatings GetForUrl.proc SocialRatings GetForUrl.Default.ResultSet0 result set](#) 15
- [proc SocialRatings GetForUser method](#) 35
- [proc SocialRatings GetForUser.ResultSet0 result set](#) 12
- [proc SocialRatings GetLastAverageSyncTime method](#) 36
- [proc SocialRatings GetRatingsForUpdate method](#) 36
- [proc SocialRatings GetRatingsForUpdate.ResultSet0 result set](#) 12
- [proc SocialRatings RefreshAverages method](#) 52
- [proc SocialRatings ReplicateForDelete method](#) 64
- [proc SocialRatings ReplicateForLive method](#) 55
- [proc SocialRatings SetLastAverageSyncTime method](#) 37
- [proc SocialTags Add method](#) 37
- [proc SocialTags Count method](#) 38
- [proc SocialTags Delete method](#) 39
- [proc SocialTags DeleteForUser method](#) 40
- [proc SocialTags GetByTime method](#) 56
- [proc SocialTags GetByTime.ResultSet0 result set \(section 2.2.4.18 17, section 2.2.4.18 17, section 2.2.4.19 17\)](#)
- [proc SocialTags GetChangedUsers method](#) 57
- [proc SocialTags GetDeletes method](#) 58
- [proc SocialTags GetDeletes.ResultSet0 result set](#) 17
- [proc SocialTags GetForReplicationByTime method](#) 58
- [proc SocialTags GetForReplicationByTime.ResultSet0 result set](#) 18
- [proc SocialTags GetForReplicationByUser method](#) 59

[proc_SocialTags_GetForReplicationByUser.ResultSet0 result set](#) 18
[proc_SocialTags_GetForTerms.method](#) 40
[proc_SocialTags_GetForTerms.ResultSet0 result set](#) 13
[proc_SocialTags_GetForUrl.method](#) 41
[proc_SocialTags_GetForUrlByTime.method](#) 60
[proc_SocialTags_GetForUrlByUser.method](#) 59
[proc_SocialTags_GetForUser.method](#) 42
[proc_SocialTags_GetForUser.ResultSet0 result set](#) 13
[proc_SocialTags_GetTerms.method](#) 43
[proc_SocialTags_GetTerms.ResultSet0 result set](#) 14
[proc_SocialTags_GetTermsForUrl.method](#) 43
[proc_SocialTags_GetTermsForUrl.proc_SocialTags_GetTermsForUrl.Default.ResultSet0 result set](#) 15
[proc_SocialTags_GetTermsForUrlBase.method](#) 61
[proc_SocialTags_GetTermsForUser.method](#) 44
[proc_SocialTags_GetTermsForUser.proc_SocialTags_GetTermsForUser.Default.ResultSet0 result set](#) 16
[proc_SocialTags_GetTermsForUser.ResultSet0 result set](#) 14
[proc_SocialTags_GetUrlCountsForTerm.method](#) 62
[proc_SocialTags_GetUrlCountsForTerm.ResultSet0 result set](#) 19
[proc_SocialTags_GetUrlsForTerm.method](#) 45
[proc_SocialTags_GetUrlsForTerm.ResultSet0 result set](#) 14
[proc_SocialTags_GetUsersForTerm.method](#) 52
[proc_SocialTags_GetUsersForTerm.ResultSet0 result set](#) 16
[proc_SocialTags_MergeTerms.method](#) 53
[proc_SocialTags_RefreshTagClouds.method](#) 53
[proc_SocialTags_ReplicateForDelete.method](#) 54
[proc_SocialTags_ReplicateForLive.method](#) 55
[proc_SocialTags_Update.method](#) 46
[Product behavior](#) 84

R

References

[informative](#) 8
[normative](#) 8
[Relationship to other protocols](#) 9
[Replicate social data example](#) 78

Result sets - messages

[proc_Social_GetUsers.ResultSet0](#) 14
[proc_SocialComments_GetForUrl.proc_SocialComments_GetForUrl.Default.ResultSet0](#) 14
[proc_SocialComments_GetForUrlBase.ResultSet0](#) 10
[proc_SocialComments_GetForUser.ResultSet0](#) 11
[proc_SocialComments_GetForUserWithTag.ResultSet0](#) 11
[proc_SocialComments_GetDeletes.ResultSet0 result set](#) 16
[proc_SocialRatings_GetForUrl.proc_SocialRatings_GetForUrl.Default.ResultSet0](#) 15
[proc_SocialRatings_GetForUser.ResultSet0](#) 12
[proc_SocialRatings_GetRatingsForUpdate.ResultSet0](#) 12

[proc_SocialTags_GetDeletes.ResultSet0](#) 17
[proc_SocialTags_GetForTerms.ResultSet0](#) 13
[proc_SocialTags_GetForUser.ResultSet0](#) 13
[proc_SocialTags_GetTerms.ResultSet0](#) 14
[proc_SocialTags_GetTermsForUrl.proc_SocialTags_GetTermsForUrl.Default.ResultSet0](#) 15
[proc_SocialTags_GetTermsForUser.proc_SocialTags_GetTermsForUser.Default.ResultSet0](#) 16
[proc_SocialTags_GetTermsForUser.ResultSet0](#) 14
[proc_SocialTags_GetUrlsForTerm.ResultSet0](#) 14
[proc_SocialTags_GetUsersForTerm.ResultSet0](#) 16

Result sets - messages

[proc_SocialRatings_GetByTime.ResultSet0](#) 19
[proc_SocialRatings_GetDeletes.ResultSet0](#) 20
[proc_SocialTags_GetByTime.ResultSet0](#) ([section 2.2.4.18](#) 17, [section 2.2.4.19](#) 17)
[proc_SocialTags_GetForReplicationByTime.ResultSet0](#) 18
[proc_SocialTags_GetForReplicationByUser.ResultSet0](#) 18
[proc_SocialTags_GetUrlCountsForTerm.ResultSet0](#) 19

S

Security

[implementer considerations](#) 83
[parameter index](#) 83

Sequencing rules

[client](#) 71
[server](#) 23

Server

[abstract data model](#) 22
[details](#) 22
[higher-layer triggered events](#) 22
[initialization](#) 22
[local events](#) 70
[message processing](#) 23
[proc_Admin_CompactChangeLogs.method](#) 66
[proc_Admin_DeletePartition.method](#) 23
[proc_Admin_RemoveSocialItems.method](#) 65
[proc_Admin_SetupPartition.method](#) 23
[proc_Social_AddSecurityTrimUrl.method](#) 69
[proc_Social_GetSecurityTrimmerEnabled.method](#) 68
[proc_Social_GetSecurityTrimUrls.method](#) 68
[proc_Social_GetUsers.method](#) 47
[proc_Social_RemoveSecurityTrimUrl.method](#) 67
[proc_Social_SetSecurityTrimmerEnabled.method](#) 67
[proc_SocialComments_Add.method](#) 23
[proc_SocialComments_Count.method](#) 25
[proc_SocialComments_Delete.method](#) 25
[proc_SocialComments_Get.method](#) 26
[proc_SocialComments_GetByTime.method](#) 47
[proc_SocialComments_GetDeletes.method](#) 48
[proc_SocialComments_GetForReplicationByTime.method](#) 48
[proc_SocialComments_GetForReplicationByUser.method](#) 49
[proc_SocialComments_GetForUrl.method](#) 27

[proc SocialComments GetForUrlByTime method](#) 49
[proc SocialComments GetForUser method](#) 28
[proc SocialComments GetForUserWithTag method](#) 29
[proc SocialComments ReplicateForDelete method](#) 50
[proc SocialComments ReplicateForLive method](#) 51
[proc SocialComments Update method](#) 29
[proc SocialData BulkDeleteForUser method](#) 66
[proc SocialData MergeSocialNotes method](#) 69
[proc SocialData MigrateSocialDataUrl method](#) 70
[proc SocialRatings Add method](#) 30
[proc SocialRatings Count method](#) 31
[proc SocialRatings Delete method](#) 32
[proc SocialRatings DeleteForUser method](#) 32
[proc SocialRatings Get method](#) 33
[proc SocialRatings GetAverage method](#) 33
[proc SocialRatings GetByTime method](#) 62
[proc SocialRatings GetDeletes method](#) 63
[proc SocialRatings GetForReplicationByTime method](#) 63
[proc SocialRatings GetForReplicationByUser method](#) 64
[proc SocialRatings GetForUrl method](#) 35
[proc SocialRatings GetForUser method](#) 35
[proc SocialRatings GetLastAverageSyncTime method](#) 36
[proc SocialRatings GetRatingsForUpdate method](#) 36
[proc SocialRatings RefreshAverages method](#) 52
[proc SocialRatings ReplicateForDelete method](#) 64
[proc SocialRatings ReplicateForLive method](#) 55
[proc SocialRatings SetLastAverageSyncTime method](#) 37
[proc SocialTags Add method](#) 37
[proc SocialTags Count method](#) 38
[proc SocialTags Delete method](#) 39
[proc SocialTags DeleteForUser method](#) 40
[proc SocialTags GetByTime method](#) 56
[proc SocialTags GetChangedUsers method](#) 57
[proc SocialTags GetDeletes method](#) 58
[proc SocialTags GetForReplicationByTime method](#) 58
[proc SocialTags GetForReplicationByUser method](#) 59
[proc SocialTags GetForTerms method](#) 40
[proc SocialTags GetForUrl method](#) 41
[proc SocialTags GetForUrlByTime method](#) 60
[proc SocialTags GetForUrlByUser method](#) 59
[proc SocialTags GetForUser method](#) 42
[proc SocialTags GetTerms method](#) 43
[proc SocialTags GetTermsForUrl method](#) 43
[proc SocialTags GetTermsForUrlBase method](#) 61
[proc SocialTags GetTermsForUser method](#) 44
[proc SocialTags GetUrlCountsForTerm method](#) 62
[proc SocialTags GetUrlsForTerm method](#) 45
[proc SocialTags GetUsersForTerm method](#) 52

[proc SocialTags MergeTerms method](#) 53
[proc SocialTags RefreshTagClouds method](#) 53
[proc SocialTags ReplicateForDelete method](#) 54
[proc SocialTags ReplicateForLive method](#) 55
[proc SocialTags Update method](#) 46
[sequencing rules](#) 23
[social comments \(notes\)](#) 22
[social ratings](#) 22
[social tags](#) 22
[timer events](#) 70
[timers](#) 22
 Simple data types
 [overview](#) 10
[Simple types - overview](#) 20
[Social comments \(notes\)](#) 22
[Social ratings](#) 22
[Social tags](#) 22
[Standards assignments](#) 9
 Structures
 [binary](#) 10
 [table and view](#) 20
 [XML](#) 20

T

[Table structures - overview](#) 20
 Timer events
 [client](#) 71
 [server](#) 70
 Timers
 [client](#) 71
 [server](#) 22
[Tracking changes](#) 85
[Transport](#) 10
 Triggered events - higher-layer
 [client](#) 71
 [server](#) 22
 Types
 [complex](#) 20
 [simple](#) 20

V

[Vendor-extensible fields](#) 9
[Versioning](#) 9
[View structures - overview](#) 20

X

[XML structures](#) 20