

# [MS-WSSEUX2]: Windows SharePoint Services: Content Database End-User Experience Communications Version 2 Protocol Specification

---

## Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft [Open Specification Promise](#) or the [Community Promise](#). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting [iplg@microsoft.com](mailto:iplg@microsoft.com).
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

**Reservation of Rights.** All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

**Tools.** The Open Specifications do not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them. Certain Open Specifications are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

## 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	Minor	Updated the technical content
03/31/2010	1.01	Editorial	Revised and edited the technical content
04/30/2010	1.02	Minor	Updated the technical content
06/07/2010	1.03	Editorial	Revised and edited the technical content
06/29/2010	1.04	Editorial	Changed language and formatting in the technical content.
07/23/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
09/27/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
06/10/2011	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
01/20/2012	1.5	Minor	Clarified the meaning of the technical content.
04/11/2012	1.5	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	1.5	No change	No changes to the meaning, language, or formatting of the technical content.

# Table of Contents

<b>1 Introduction</b>	<b>8</b>
1.1 Glossary	8
1.2 References	10
1.2.1 Normative References	10
1.2.2 Informative References	11
1.3 Protocol Overview (Synopsis)	11
1.3.1 Alerts	11
1.3.2 E-mail Integration	11
1.3.3 HTML Translate Cache	12
1.3.4 Meetings	12
1.3.5 Multilingual User Interface	12
1.3.6 Navigation Structure	12
1.3.7 Tree View	12
1.3.8 Web Discussions	12
1.4 Relationship to Other Protocols	12
1.5 Prerequisites/Preconditions	13
1.6 Applicability Statement	13
1.7 Versioning and Capability Negotiation	13
1.8 Vendor-Extensible Fields	13
1.9 Standards Assignments	13
<b>2 Messages</b>	<b>14</b>
2.1 Transport	14
2.2 Common Data Types	14
2.2.1 Simple Data Types and Enumerations	14
2.2.1.1 Alert Delivery Channel	14
2.2.1.2 Alert Status	14
2.2.1.3 Alert	14
2.2.1.4 Meetings Event	15
2.2.1.5 Navigation Node	15
2.2.1.6 Notification Frequency	15
2.2.1.7 RecurrenceId	15
2.2.1.8 Sequence	15
2.2.1.9 UID	16
2.2.1.10 User Resource	16
2.2.2 Bit Fields and Flag Structures	16
2.2.2.1 Alert Event Type	16
2.2.2.2 Special Alert Flags	16
2.2.2.3 Web Discussion Comment Status Flags	17
2.2.3 Binary Structures	17
2.2.3.1 Cached Nav Data	17
2.2.4 Result Sets	18
2.2.4.1 EmailAliases Result Set	18
2.2.4.2 Nav ACLs Result Set	18
2.2.4.3 Nav Data Result Set	19
2.2.4.4 Nav Children Result Set	20
2.2.5 Tables and Views	20
2.2.5.1 AllUserData Table	20
2.2.5.2 UserInfo Table	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 .....	20
2.2.6.6	Groups .....	20
2.2.6.7	Attribute Groups.....	20
<b>3</b>	<b>Protocol Details.....</b>	<b>21</b>
3.1	Back-end Database Server Details .....	21
3.1.1	Abstract Data Model .....	21
3.1.1.1	Alerts.....	21
3.1.1.2	HTML Translate Cache.....	21
3.1.1.3	Meetings .....	22
3.1.1.4	Multilingual User Interface .....	22
3.1.1.5	Navigation Structure .....	23
3.1.1.6	TreeView .....	24
3.1.1.7	Web Discussions.....	24
3.1.2	Timers .....	24
3.1.3	Initialization .....	24
3.1.4	Higher-Layer Triggered Events.....	24
3.1.5	Message Processing Events and Sequencing Rules.....	24
3.1.5.1	proc_AddDocComment .....	24
3.1.5.1.1	AddDocComment Result Set .....	26
3.1.5.2	proc_AddSubscription .....	26
3.1.5.3	proc_AddUserResource.....	29
3.1.5.4	proc_CheckMeetingInstance.....	30
3.1.5.4.1	Meeting Event Result Set .....	31
3.1.5.5	proc_CheckNavStructContainsPage .....	32
3.1.5.6	proc_DeleteSubscription .....	33
3.1.5.7	proc_DeleteSubscriptionJunctionEntries.....	33
3.1.5.8	proc_EnumDoclibsFileDlg.....	38
3.1.5.8.1	EnumerationWithTemplates Result Set.....	39
3.1.5.8.2	EnumerationWithoutTemplates Result Set .....	39
3.1.5.9	proc_EnumEmailAliases .....	40
3.1.5.10	proc_EnumEmailAliasesBySite .....	40
3.1.5.11	proc_EnumResourcesAtScope.....	40
3.1.5.11.1	EnumResAtScopeResultSet Result Set.....	41
3.1.5.12	proc_EnumResourceValuesForAllLangs .....	41
3.1.5.12.1	EnumResValueLangResultSet Result Set.....	42
3.1.5.13	proc_EnumSubscribedSites .....	42
3.1.5.13.1	SubscribedSites Result Set .....	42
3.1.5.14	proc_GetAlertsSqmData.....	43
3.1.5.14.1	AlertsSqmData Result Set.....	43
3.1.5.15	proc_GetDefaultMtgInstance .....	43
3.1.5.15.1	Default Meeting Instance Result Set .....	44
3.1.5.16	proc_GetDocComments .....	45
3.1.5.16.1	GetDocComments Result Set .....	45
3.1.5.17	proc_GetEventDataAndSubscriptionFilters.....	46
3.1.5.17.1	Events Result Set .....	46
3.1.5.17.2	Subscriptions Result Set.....	47
3.1.5.17.3	UTCTime Result Set.....	49
3.1.5.18	proc_GetExceptionIDs .....	49

3.1.5.18.1	Exceptions Result Set .....	50
3.1.5.19	proc_GetFutureExceptionIDsForUID .....	50
3.1.5.19.1	Exceptions Result Set .....	51
3.1.5.20	proc_GetItemCountPerInstance .....	52
3.1.5.21	proc_GetListAndChildrenNSInfo .....	52
3.1.5.21.1	ListRootFolderNSInfo Result Set .....	53
3.1.5.21.2	ChildFoldersNSInfo Result Set .....	54
3.1.5.22	proc_GetMatchingSubscriptionsData.....	55
3.1.5.22.1	MatchingSubscriptionsData Result Set.....	59
3.1.5.23	proc_GetMeetingInstanceDataForICal.....	61
3.1.5.24	proc_GetRecurrenceSeriesData.....	62
3.1.5.25	proc_GetResourcesAtScope .....	64
3.1.5.25.1	Get Resources Scope Nvarchar Result Set .....	64
3.1.5.25.2	Get Resources Scope Ntext Result Set .....	65
3.1.5.26	proc_GetVersionIndependentMetaInfo .....	65
3.1.5.26.1	VersionMetaInfo Result Set.....	66
3.1.5.27	proc_GetWebAndChildrenNSInfo .....	66
3.1.5.27.1	SiteNSInfo Result Set .....	67
3.1.5.27.2	ListNSInfo Result Set .....	68
3.1.5.28	proc_GetWebComments .....	69
3.1.5.28.1	GetWebComments Result Set .....	69
3.1.5.29	proc_GetWebNavAcls .....	70
3.1.5.30	proc_GetWebNavStruct .....	71
3.1.5.30.1	Site Scope and CachedDataVersion Result Set .....	71
3.1.5.30.2	Nav ACLs Result Set .....	71
3.1.5.30.3	Nav Data Result Set.....	71
3.1.5.31	proc_GetWebNavStructNodeByIds .....	73
3.1.5.32	proc_GetWebSubscriptions.....	73
3.1.5.32.1	WebSubscriptions Result Set.....	74
3.1.5.33	proc_GetWebSubscriptionsForBackup .....	75
3.1.5.33.1	WebSubscriptionsForBackupResult Set .....	76
3.1.5.34	proc_GetWebSubscriptionsUniqueUsers .....	77
3.1.5.34.1	WebSubscriptionsUniqueUsers Result Set .....	77
3.1.5.35	proc_HandleMtgRecurPatternChange .....	77
3.1.5.36	proc_HTCreatRow .....	78
3.1.5.37	proc_HTGetFile.....	79
3.1.5.37.1	File Result Set.....	80
3.1.5.38	proc_HTGuidFromOrig .....	80
3.1.5.38.1	Translation Name Result Set .....	81
3.1.5.39	proc_InstantiateMtgSeriesOccurrence.....	81
3.1.5.40	proc_ListDocsHavingComments .....	83
3.1.5.40.1	ListDocsHavingComments Result Set .....	83
3.1.5.41	proc_MatchSchedSubscriptions.....	84
3.1.5.41.1	SchedSubscriptions Result Set .....	85
3.1.5.42	proc_ModifySubscription .....	87
3.1.5.43	proc_MoveDataToOccurrence .....	89
3.1.5.44	proc_NavStructAddNewNodeByDocId .....	91
3.1.5.44.1	Nav Data Result Set.....	93
3.1.5.44.2	Nav Children Result Set.....	93
3.1.5.45	proc_NavStructAddNewNodeByUrl .....	93
3.1.5.45.1	Nav Data Result Set.....	95
3.1.5.45.2	Nav Children Result Set.....	95
3.1.5.46	proc_NavStructAllocateEidBlockWebId .....	95

3.1.5.47	proc_NavStructClear .....	96
3.1.5.48	proc_NavStructDeleteNodeByIds .....	96
3.1.5.49	proc_NavStructLogChangesAndUpdateSiteChangedTime .....	97
3.1.5.50	proc_NavStructMoveNode .....	97
3.1.5.50.1	Nav Data Result Set.....	99
3.1.5.50.2	Nav Children Result Set.....	99
3.1.5.51	proc_OrphanRecurringEventExceptions.....	99
3.1.5.52	proc_ProcessDelMtgAttendeeListItem .....	100
3.1.5.52.1	Site Audit Mask Result Set.....	101
3.1.5.53	proc_PutWebNavStructNode.....	101
3.1.5.53.1	Old Name Result Set.....	103
3.1.5.53.2	Nav Data Result Set.....	103
3.1.5.53.3	Nav Children Result Set.....	104
3.1.5.54	proc_RelinkMeeting .....	104
3.1.5.55	proc_ResolveWikiLinkFile .....	104
3.1.5.55.1	Resolve Wiki Link File Result Set .....	105
3.1.5.56	proc_ResolveWikiLinkItem .....	105
3.1.5.56.1	Resolve Wiki Link Item Result Set.....	105
3.1.5.57	proc_SetCommentAttribs .....	106
3.1.5.58	proc_SetSecondaryResourceAtScope .....	107
3.1.5.59	proc_SetVersionIndependentMetaInfo .....	107
3.1.5.60	proc_SetWebNavStructInheritance .....	108
3.1.5.61	proc_UnOrphanMtgOccurrence .....	109
3.1.5.62	proc_UpdateCachedNav .....	110
3.1.5.63	proc_UpdateItemJunctionsCurrentVersion .....	111
3.1.5.64	proc_UpdateListNavNode .....	112
3.1.5.65	proc_UpdateSchedSubscriptionTimes .....	112
3.1.5.66	proc_UpdateUserResource .....	113
3.1.6	Timer Events .....	114
3.1.7	Other Local Events .....	114
3.2	WSSEUX Client Details .....	114
3.2.1	Abstract Data Model .....	114
3.2.2	Timers .....	114
3.2.3	Initialization .....	115
3.2.4	Higher-Layer Triggered Events.....	115
3.2.4.1	Fetch HTML Translate Cache Item.....	115
3.2.4.2	CreateNavigationNodeByDocId .....	116
3.2.4.3	CreateNavigationNodeByUrl .....	116
3.2.4.4	AddAndUpdateNavigationNode .....	117
3.2.4.5	MoveNavigationNode.....	117
3.2.5	Message Processing Events and Sequencing Rules.....	118
3.2.6	Timer Events .....	118
3.2.7	Other Local Events .....	118
<b>4</b>	<b>Protocol Examples.....</b>	<b>119</b>
4.1	TreeView Operations .....	119
4.2	HTML Translate Cache .....	119
4.3	Web Discussions Operations.....	119
4.3.1	Add a Comment .....	119
4.3.2	Reply to a Comment .....	119
4.3.3	Edit a Comment .....	119
4.3.4	Delete a Comment .....	120
4.3.5	Enumerate All Comments in a Document.....	120

4.4	Navigation Structure .....	120
4.4.1	Create a Navigation Node to a URL .....	120
4.4.2	Move a Navigation Node.....	120
4.4.3	Enumerate Navigation Nodes in a Site .....	120
4.5	User Resource String.....	120
4.5.1	Add a User Resource String .....	120
4.5.2	Update a User Resource String.....	120
<b>5</b>	<b>Security.....</b>	<b>121</b>
5.1	Security Considerations for Implementers.....	121
5.2	Index of Security Parameters .....	121
<b>6</b>	<b>Appendix A: Product Behavior.....</b>	<b>122</b>
<b>7</b>	<b>Change Tracking.....</b>	<b>123</b>
<b>8</b>	<b>Index .....</b>	<b>124</b>

# 1 Introduction

This document specifies the Windows SharePoint Services: Content Database End-User Experience Communications Version 2 Protocol. This protocol specifies communications between servers that support various end-user experiences, such as online discussions, meeting arrangements, Web-site navigation, and alerts about changed data.

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

## 1.1 Glossary

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

- access control list (ACL)**
- anonymous user**
- Coordinated Universal Time (UTC)**
- GUID**
- language code identifier (LCID)**
- object**
- resource**
- Unicode**
- unique identifier (UID)**
- XML**

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

- 12-hour clock notation**
- 24-hour clock notation**
- alert**
- alert subscription**
- alert template**
- always notify alert**
- audit flag**
- back-end database server**
- base type**
- calendar type**
- CAML**
- Central Administration site**
- change log**
- checked out**
- Collaborative Application Markup Language (CAML)**
- collation**
- collation order**
- column**
- content database**
- content type**
- current user**
- datetime**
- default language**
- default list view**
- directory name**
- dirty**



**display name**  
**document**  
**document identifier**  
**document library**  
**document store type**  
**document template**  
**e-mail alias**  
**embedded image**  
**empty GUID**  
**empty string**  
**event**  
**field**  
**file**  
**folder**  
**form**  
**front-end Web server**  
**home page**  
**HTML translate cache**  
**Hypertext Markup Language (HTML)**  
**iCalendar**  
**inheritable navigation structure**  
**item**  
**item identifier**  
**leaf name**  
**level**  
**link bar**  
**links list**  
**list**  
**list identifier**  
**list item**  
**list item identifier**  
**list template**  
**list view page**  
**locale**  
**locked**  
**login name**  
**lookup field**  
**meeting instance**  
**Meeting Workspace site**  
**metadict**  
**navigation node**  
**navigation node element identifier**  
**navigation structure**  
**navigational parent site**  
**parent site**  
**permission**  
**permissions mask**  
**published**  
**query**  
**read-only mode**  
**record**  
**recurrence list**  
**recurrence rule**  
**request identifier**  
**result set**

**return code**  
**root folder**  
**row**  
**scope identifier**  
**security principal**  
**security scope**  
**sequence number**  
**server-relative URL**  
**site**  
**site collection**  
**site collection identifier**  
**site content type**  
**site definition**  
**site identifier**  
**site template**  
**SQL collation name**  
**stored procedure**  
**store-relative form**  
**store-relative URL**  
**Structured Query Language (SQL)**  
**subscription**  
**subsite**  
**system alert**  
**time zone**  
**Transact-Structured Query Language (T-SQL)**  
**Uniform Resource Locator (URL)**  
**user identifier**  
**user name**  
**Web discussion comment**  
**Windows collation name**  
**XML fragment**

The following terms are specific to this document:

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

## 1.2 References

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

### 1.2.1 Normative References

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

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

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

[MS-FPSE] Microsoft Corporation, "[FrontPage Server Extensions Remote Protocol Specification](#)".

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

[MS-WSSDLIM] Microsoft Corporation, "[Windows SharePoint Services: Content Database Document and List Item Management Communications Protocol Specification](#)".

[MS-WSSDM2] Microsoft Corporation, "[Windows SharePoint Services: Content Database Data Migration Communications Version 2 Protocol Specification](#)".

[MS-WSSFO2] Microsoft Corporation, "[Windows SharePoint Services \(WSS\): File Operations Database Communications Version 2 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>

[RFC2445] Dawson, F., and Stenerson, D., "Internet Calendaring and Scheduling Core Object Specification (iCalendar)", RFC 2445, November 1998, <http://www.rfc-editor.org/rfc/rfc2445.txt>

## 1.2.2 Informative References

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

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

[MS-OUTSPS] Microsoft Corporation, "[Lists Client Sync Protocol Specification](#)".

## 1.3 Protocol Overview (Synopsis)

This protocol specifies the communication between the **front-end Web server** and the back-end **content database** used in user interaction with the server. The content database stores the data associated with the **lists (1)** and **sites (2)**. The user sends a request for a certain operation or data to the front-end Web server. The front-end Web server then communicates with the content database to perform this action. This client-to-server protocol uses the [\[MS-TDS\]](#) as its transport between the front-end Web server, acting as a client, and the database, acting as a server.

### 1.3.1 Alerts

The user can request notifications for changes to the content in the server. When the requested **document** (or other server **object (1)**) changes, the front-end Web server sends notifications to the user using the e-mail service. The request for notifications, called **alert subscriptions**, are stored in the server. The front-end Web server receives request to enumerate the **alerts (1)** for a user, site (2) or **list item** or modify the alerts (1). The front-end Web server uses the protocol to retrieve the data from the **back-end database server** and send it to the user. The front-end Web server tries to match alerts (1) with documents (or other server objects (1)) that have changes and sends notifications to users. The notifications can be sent immediately or batched together and sent at a particular interval as requested by the user.

### 1.3.2 E-mail Integration

Users can associate a list (1) with an **e-mail alias**. The protocol enables the front-end Web server to enumerate all the e-mail aliases associated with a site (2) or all the e-mail aliases stored in the content database.

### 1.3.3 HTML Translate Cache

The **HTML translate cache** is used to store Web-viewable versions of files in the back-end database server. For example, it could be used to store an HTML version of a binary **file**.

### 1.3.4 Meetings

Users can create calendar events and meetings in a site (2). They are created as list items in a calendar events list (1). Users can enumerate and update these items with new information. They can associate a **Meeting Workspace site** with each meeting. A Meeting Workspace site is a convenient and centralized place for project collaboration and meeting proceedings. For example, after a meeting, results are **published** on the Meeting Workspace site. The front-end Web server updates the associated Meeting Workspace site whenever the user updates the meeting. Recurring meetings are meetings that occur periodically over a period of time. There can be exceptions to these meetings. For example, a meeting can occur every week on Monday at 10 A.M. except on the 15<sup>th</sup> of each month when it occurs at 11 A.M.

### 1.3.5 Multilingual User Interface

Users can view and edit sites (2) in their preferred language with a quick switch of the language settings. Users can also provide translations for elements created by them, such as lists (1), **content types**, and **navigation nodes**. The **resources** corresponding to the translations for the user created elements are stored in the back-end database server. Available languages are limited by the language packs that are installed.

### 1.3.6 Navigation Structure

The **navigation structure** is a hierarchical representation of related **Uniform Resource Locators (URLs)**. Every site (2) has one associated navigation structure. The purpose of the navigation structure is to define and render links to related URLs. Each element in the hierarchy is a navigation node. A navigation node can be based either on the files and **folders** in the **site collection** or on literal URL strings. The user can create, enumerate, modify, delete, and organize these navigation nodes in the navigation structure.

### 1.3.7 Tree View

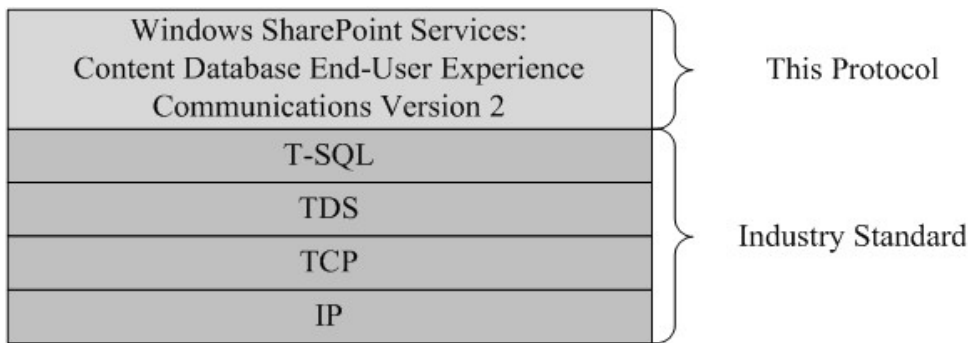
A site (2) can contain a number of **subsites**, lists (1), and folders. A tree view is a graphical user interface that provides a hierarchical view of the information. A tree view provides a top-down enumeration of the content in the site (2). Each node can have several subitems. They can be expanded to reveal the subitems and collapsed to hide the subitems. The user can navigate to the containing subsites, lists (1), and folders within a node.

### 1.3.8 Web Discussions

An important part of collaboration is discussing and reviewing the contents of a document. To enable discussions, the front-end Web server provides the ability for the user to comment on documents. Users can view the comments, modify them, or delete them. The comments are stored in the back-end database server; the front-end Web server uses the protocol to create and manipulate the comments as requested by the user.

## 1.4 Relationship to Other Protocols

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



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

## 1.5 Prerequisites/Preconditions

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

## 1.7 Versioning and Capability Negotiation

- **Security and Authentication Methods:** This protocol supports the SSPI and SQL Authentication with the Protocol Server role specified in [\[MS-TDS\]](#).

## 1.8 Vendor-Extensible Fields

None.

## 1.9 Standards Assignments

None.

## 2 Messages

### 2.1 Transport

[\[MS-TDS\]](#) is the transport protocol used to call the stored procedures, **query SQL** tables, return result 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

This section contains common simple data types and enumerations used by this protocol.

##### 2.2.1.1 Alert Delivery Channel

An **integer** value that specifies the delivery channel of the alert subscription. It MUST be one of the following:

Value	Description
"0"	None.
"1"	Delivered as e-mail.
"2"	Delivered as SMS message.

##### 2.2.1.2 Alert Status

An **integer** value that specifies the status of the alert subscription. It MUST be one of the following:

Value	Description
"0"	On
"1"	Off
"2"	Error

##### 2.2.1.3 Alert

An **integer** value that specifies the type of alert subscription. It MUST be one of the following:

Value	Description
"0"	List (1) alert (1)
"1"	<b>Item</b> alert (1)
"2"	Custom alert (1)

### 2.2.1.4 Meetings Event

An **integer** value that describes the type of calendar event or meeting. Its value MUST be one of the following:

Value	Description
"0"	A single nonrecurring calendar event.
"1"	A recurring calendar event.
"2"	An exception from the recurrence series.
3	An exception that has been deleted.
4	An exception from the recurrence series <a href="#">&lt;1&gt;</a> .
5	A single instance of a meetings workspace.

### 2.2.1.5 Navigation Node

An **integer** value that defines whether the navigation node in the navigation structure is based on the document identifier of a document within the site collection or on a literal URL. It MUST be one of the following values:

Value	Description
"0"	The navigation node is based on a document identifier.
"1"	The navigation node is based on a URL.

### 2.2.1.6 Notification Frequency

This specifies the frequency of the alert (1) notifications. It is an **integer** value and MUST be listed in the following table:

Value	Description
"0"	Immediate alert (1): Notification is fired immediately after a change.
"1"	Daily alert (1): Notification is fired once per day.
"2"	Weekly alert (1): Notification is fired once per week.

### 2.2.1.7 RecurrenceId

This element contains an identifier specifying a particular instance of a recurring meeting.

### 2.2.1.8 Sequence

This element contains a sequence or revision number of a **meeting instance**. This is defined in [\[RFC2445\]](#).

### 2.2.1.9 UID

This element contains a unique identifier specifying a meeting. This MUST be a valid **unique identifier (UID)** string as used by calendaring objects (1) and defined in [\[RFC2445\]](#). For a recurring meeting, each meeting instance shares a common UID. To distinguish between these meeting instances, each recurring meeting instance is also assigned a recurrence identifier. This is typically the date on which that meeting instance occurs in the series that was calculated with the **time zone** of the meeting organizer.

### 2.2.1.10 User Resource

An **integer** value that specifies whether the user resource has a single line of text or multiple lines of text. It MUST be one of the following:

Value	Description
"0"	Single line text.
"1"	Multiple lines of text

## 2.2.2 Bit Fields and Flag Structures

### 2.2.2.1 Alert Event Type

The **integer** mask for the types of **events (2)** on which to fire the alert (1) notification. It MUST be one of the following:

Value	Description
"-1"	All event (2) types.
"1"	Items are added.
"2"	Items are modified.
"4"	Items are deleted.
"9"	Items are restored.
"4080"	Discussion related.

### 2.2.2.2 Special Alert Flags

This flag specifies whether the alert subscription is an **always notify alert** or a **system alert**.

Value	Description
"0x40000000"	The alert (1) is an always notify alert.
"0x20000000"	The alert (1) is a system alert.



### 2.2.2.3 Web Discussion Comment Status Flags

A 2 byte unsigned **integer** bit mask describing the status of a **Web discussion comment**, which can have one or more flags set. The values of the Web Discussion Comment Status Flags bits are specified as follows.

Value	Description
"0x00000001"	The Web discussion comment is deleted. Protocol servers MUST permanently delete this Web discussion comment if all of its child objects (1) have a Web discussion comment flags value of "0x00000001".
"0x00000002"	The Web discussion comment is closed. Protocol clients MUST disable modifications and replies to this item unless the flag "0x00000008" is also set.
"0x00000004"	The thread beginning at this Web discussion comment is closed. Protocol clients MUST disable modifications and replies to this Web discussion comment, and any of its child Web discussion comments that do not have flag value "0x00000008" set.
"0x00000008"	This Web discussion comment is active. Protocol clients MUST ignore any flag value of "0x00000002".

### 2.2.3 Binary Structures

None.

#### 2.2.3.1 Cached Nav Data

The **Cached Nav Data** structure contains the node structure in the following binary data format:

```
Navigation Node Element Identifier,  
NULL,  
Navigation Node Element Identifier of the parent Navigation Node,  
NULL,  
Name of the User Resource corresponding to this Navigation Node,  
NULL  
URL of the Navigation Node,  
NULL,  
Scope Identifier,  
NULL,  
Security Type,  
NULL,  
Document Store Type of the Document this Navigation Node points to,  
NULL
```

These identifiers are stored as **Unicode** streams.

**Security Type** is an **integer** value that defines the permissions required to see this navigation node. The following are all possible valid values:

Value	Description
"0"	This navigation node is always visible to the user.
"1"	This navigation node requires the user to have permissions to view Web pages in the site (2) that are not in a list (1) or <b>document library</b> .

Value	Description
"2"	This navigation node requires the user to have permissions to view Web pages in the site (2) that are <b>form (1)</b> or <b>list view pages</b> of a list (1) or document library.
"3"	This navigation node requires the user to have permissions to view Web pages in the site (2) that are items in a document library.

## 2.2.4 Result Sets

### 2.2.4.1 EmailAliases Result Set

The **EmailAliases** result set MUST contain e-mail aliases and associated **list identifier**, **site identifier**, and **site collection identifier**. All e-mail aliases returned MUST NOT be NULL. The **T-SQL** syntax for the result set is as follows:

```
tp_EmailAlias          nvarchar(128),
SiteId                 uniqueidentifier,
tp_WebId               uniqueidentifier,
tp_ID                  uniqueidentifier;
```

**tp\_EmailAlias:** The string name of the e-mail alias.

**SiteId:** The identifier of the **site collection** to which the e-mail alias belongs.

**tp\_WebId:** The identifier of the site (2) to which the e-mail alias belongs.

**tp\_ID:** The identifier of the list (1) to which the e-mail alias belongs.

### 2.2.4.2 Nav ACLs Result Set

The **Nav ACLs** result set returns the **access control lists (ACLs)** of all the unique **security scopes** of documents in the navigation structure of the site (2). This result set MUST be returned. The number of **rows (3)** in this result set depends on the values of *@NavParentWebId* and *@Inherited*.

Value of <i>@NavParentWebId</i>	Value of <i>@Inherited</i>	Number of rows (3) returned
NULL	"0"	One row (3) for each distinct security scope for the navigation nodes that are different from the security scope of the site (2). If there are no distinct security scopes for the navigation nodes of the site (2) other than the security scope of the site (2) itself, this result set MUST return zero rows (3).
NULL	"1"	One row (3) for each distinct security scope for the navigation nodes of the site (2) including the security scope of the site (2) itself.
NOT NULL	"0"/"1"	One row (3) for each distinct security scope for the navigation nodes of the site (2), plus one row (3) for each distinct security scope for the navigation nodes of the <b>navigation parent site</b> , plus one row (3) for the security scope of the navigational parent site.

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

ScopeId	uniqueidentifier,
Acl	varbinary(max),
AnonymousPermMask	bigint;

**ScopeId:** The identifier of the security scope.

**Acl:** The binary serialization of the **ACL** for this security scope.

**AnonymousPermMask:** The **permissions mask** that applies to an **anonymous user** of the site (2) or a user who has no specific rights for this security scope. The format is specified in [\[MS-WSSFO2\]](#) section 2.2.2.14.

### 2.2.4.3 Nav Data Result Set

The **Nav Data** result set returns information about the specified navigation node. This result set **MUST** always be returned. If the navigation node specified by *@Eid* exists for the site (2), this **MUST** return one row (3); otherwise it **MUST** return no rows (3). The T-SQL syntax for the result set is as follows:

Eid	int,
EidParent	int,
ElementType	tinyint,
{Url}	nvarchar(385),
Name	nvarchar(256),
NodeMetainfo	varbinary(max),
NonNavPage	bit,
NavSequence	bit,
ChildOfSequence	bit,
DateLastModified	datetime;

**Eid:** The **navigation node element identifier** of the navigation node.

**EidParent:** The identifier of the parent navigation node.

**ElementType:** Specifies the navigation node type (section [2.2.1.5](#)) of the navigation node.

**{Url}:** The URL to which the navigation node points. This value **MUST NOT** be NULL if the **Eid** value is not zero.

**Name:** The **display name** of the navigation node.

**NodeMetainfo:** A binary serialization of the metadata for a navigation node in **metadict** form ([\[MS-FPSE\]](#) section 2.2.2.2.11). This **MAY** be NULL.

**NonNavPage:** A bit specifying whether the navigation node is filtered out when rendering the navigation structure. If the bit is set to "1", the navigation node **SHOULD** be filtered out [<2>](#). If it is set to zero ("0"), it **MUST NOT** be filtered out.

**NavSequence:** A bit specifying whether the navigation node represents a **link bar**. If this navigation node represents a link bar, this bit **MUST** be set to "1"; otherwise it **MUST** be set to zero.

**ChildOfSequence:** A bit specifying whether the navigation node is a child navigation node of a link bar. If it is, this bit **MUST** be set to "1"; otherwise it **MUST** be set to zero.

**DateLastModified:** The time in **Coordinated Universal Time (UTC)** when the navigation node information was last modified.

#### 2.2.4.4 Nav Children Result Set

The **Nav Children** result set returns the navigation node element identifier of all the child objects (1) of the navigation node. This result set **MUST** be returned, and it **MUST** contain one row (3) for each navigation node that is a child of the specified navigation node. If the navigation node has no child objects (1), this **MUST** return no rows (3). The T-SQL syntax for the result set is as follows:

```
Eid                int;
```

**Eid:** The navigation node element identifier of the child navigation node.

### 2.2.5 Tables and Views

#### 2.2.5.1 AllUserData Table

Specified in [\[MS-WSSFO2\]](#) section 2.2.7.3.

#### 2.2.5.2 UserInfo Table

Specified in [\[MS-WSSFO2\]](#) section 2.2.7.10.

### 2.2.6 XML Structures

None.

#### 2.2.6.1 Namespaces

None.

#### 2.2.6.2 Simple Types

None.

#### 2.2.6.3 Complex Types

None.

#### 2.2.6.4 Elements

None.

#### 2.2.6.5 Attributes

None.

#### 2.2.6.6 Groups

None.

#### 2.2.6.7 Attribute Groups

None.

## 3 Protocol Details

### 3.1 Back-end Database Server Details

#### 3.1.1 Abstract Data Model

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

The descriptions, in sections 3.1.4.x, of the parameters for stored procedures and the **fields (3)** of result sets imply the kinds and types of data that are stored by the front-end Web servers and back-end database servers. In particular, note the following:

- The back-end database server stores an **integer**, for each site (2), that expresses the number of days to extend or reduce the current month in Hijri calendars.

##### 3.1.1.1 Alerts

An alert subscription is a persistent request on the server for notifications, or alerts (1), that a particular document or list (1) has been modified. The user can also request notifications only for certain kinds of modifications. For example, an alert (1) can be requested if the value of a specific field (3) changes to a particular value, or if the modifier of the item is a specific user. The alert subscription contains the information that determines when to generate a notification, and how to deliver the notification. When a document or a list item changes, the server checks whether an alert subscription is present for this change. If the server finds a matching alert subscription, then it sends an e-mail notification to the user.

Each alert subscription has a **GUID** and a user associated with it. In addition, it has the site (2), Web, list (1), and item information required to identify the object (1) for whose changes the user has requested notification. The protocol allows for enumeration of alert subscriptions associated with a user or Web. When the front-end Web server receives a request to modify or delete the alert subscription, it uses the protocol to perform the action on the back-end database server. There are two kinds of alert subscriptions: immediate and scheduled. The server sends alert (1) notifications for immediate alert subscriptions as soon as the changes occur; notifications for scheduled alert subscriptions are sent at the requested time as specified by the Notification Frequency in the alert subscription.

The changes to the objects (1) are stored as events (2) in a **change log**. The server matches the events (2) in the change log with the list (1) of immediate and scheduled alert subscriptions and sends notifications for events (2) that match alert subscriptions.

##### 3.1.1.2 HTML Translate Cache

The HTML translate cache is used to store Web-viewable versions of files. Because many files have the ability to contain other files, such as a binary file containing images, the **HTML** version of a single original file often is a collection of multiple files. The core file for a set of translated files is called the main file. The other files, such as an **embedded image** file, are supporting files.

Files are retrieved from the HTML translate cache by specifying the **directory name** and **leaf name** of the original file as well as the name of the translated file. The name of the translation of the main file can be retrieved using **proc\_HTGuidFromOrig** (section [3.1.5.38](#)). The names of the translated

versions of the supporting files cannot be retrieved from the HTML translate cache and so they MUST either be recorded or they MUST be derivable from the requesting URL.

Here is an example of a system where a translated name for a supporting file can be derived from the URL. The URL to the main file is

<Any Unique Path>/~/<Main File TransName>

Because the name of the translated main file can be retrieved using **proc\_HTGuidFromOrig**, the knowledge of the path to the original file is sufficient to render that URL. The name for the translated main file is <GUID>/HtmlView.htm.

The GUID is generated when the translated files are placed into the HTML translate cache, but is not recorded by the user of the cache. The name for the translated supporting file is

<GUID>/<Subfolder Name>/<Supporting File Leaf Name>

Subfolder Name is a subfolder name supplied by the component that performed the translation to HTML, and all links to supporting files within the main file are relative links of the form (1) <Subfolder Name>/<Supporting File Leaf Name>.

The browser, when it sees a link to a supporting file while parsing the main file, will use the relative link within the main file to construct a request to the server with the URL.

<Any Unique Path>/~/<Main File TransName>/<Subfolder Name>/Supporting File Leaf Name>

Looking at the preceding example, one can see that the name for the translated supporting file is the part of the path after the tilde. Thus, all the files can be retrieved using **proc\_HTGuidFromOrig** and **proc\_HTGetFile** (section [3.1.5.37](#)) without need for the front-end Web server recording the name of any of the translated files.

### 3.1.1.3 Meetings

Users can create calendar events and meetings in a site (2). They are stored as list items inside a list that is of **base type Events**. The meeting time, location, the title of the meeting, attendees and other information pertaining to the meeting are stored in the list item in the back-end database server. Each meeting is uniquely identified by a UID. Meetings can also be created as recurring meetings. The recurrence series is specified by the recurrence data that is expressed in **XML** format ([\[MS-OUTSPS\]](#), section [2.2.4.2](#) and section [2.2.4.5](#)). The recurring meeting is stored as a single item in the containing list (1), and it contains the recurrence data. Each instance of a recurring meeting is identified by the [RecurrenceId](#). Exceptions can be created for recurring meetings. An exception is a single appointment that overrides one instance of a recurring appointment. The beginning date and time of the overridden instance is the replacement date-time of the exception. Any property of a recurrence that is missing from an exception is assumed to have the same value as other instances of the recurrence. This means an exception with no location to a recurrence with location="xyz" has a location "xyz". All properties of an exception override properties of a recurrence. Exceptions are in the time zone of the recurrence they belong to. Each exception is stored separately in the back-end database server. A Meeting Workspace site can be associated with each meeting. A Meeting Workspace site can be associated with multiple single instance meetings or one recurring meeting.

### 3.1.1.4 Multilingual User Interface

Multilingual User Interface allows users to view and edit sites (2) in their preferred language. Users can also provide translations for customized and new application content, such as list (1) title, list (1) description, content type name, and navigation node name. These translations are called user

resources and are stored in back-end database server. User resources for elements associated with a specific list (1), such as list (1) title, and list (1) field, are considered list (1) scoped while those not associated with any list (1), such as navigation nodes and **site content types**, are site (2) scoped.

When rendering a Web page in the user's preferred language that is different from the site's **default language**, all user resources necessary for rendering that page are fetched by making calls to **proc\_EnumResourcesAtScope** (section [3.1.5.11](#)) for each site (2) and list (1) that have information about the Web page.

The translations for user resources for a site (2) and all lists (1) in the site (2) can be done in bulk. All such user resources are fetched using **proc\_GetResourcesAtScope** (section [3.1.5.25](#)) and the translated values are set using **proc\_SetSecondaryResourceAtScope** (section [3.1.5.58](#))

### 3.1.1.5 Navigation Structure

The navigation structure is a hierarchical representation of related URLs. Each element in the hierarchy is a navigation node. A navigation node can be based on either the files and folders in the site collection or the literal URL strings. Each navigation node MUST have a unique navigation node element identifier for itself, a navigation node element identifier pointing to its parent in the hierarchy, a display name, a bit representing whether it is a link bar, and a bit representing whether it SHOULD be hidden when rendering. A navigation node can also have metadata associated with it.

Every site (2) MUST have a navigation node with its element identifier as zero. This node MUST be the root of the hierarchical representation.

When adding navigation nodes, temporary identifiers for navigation node elements can be used. Identifiers for navigation node elements from 1 to 999 MUST be considered temporary by the back-end database server. All calls to add, move, or update navigation nodes using temporary identifiers for navigation node elements MUST contain a base value. This MUST be obtained by a call to `proc_NavStructAllocateEidBlockWebId` (section [3.1.5.46](#)). The base value MUST be used to generate valid identifiers for navigation nodes from the temporary identifiers for navigation node elements.

Calls from the front-end Web server to the back-end database server to add, move, update, and delete navigation nodes can be batched together into a single transaction. In that case, a single call to `proc_NavStructAllocateEidBlockWebId` (section [3.1.5.46](#)) MUST be made first, and the `@@EidBase` output parameter MUST be used as the base value for all subsequent calls in the batch.

The navigation node with navigation node element identifier "1000" MUST NOT be a child of a link bar and MUST point to the **home page** of the site (2). If a new navigation node pointing to the home page of the site (2) and of navigation node type zero is not being added as a child of a link bar, its navigation node element identifier MUST be changed to "1000". If this add is being done as part of a batch, the navigation node element identifier that would have been generated for this Node MUST be returned as the `@@EidHome` output parameter for **proc\_NavStructAddNewNodeByDocId** (section [3.1.5.44](#)) and **proc\_NavStructAddNewNodeByUrl** (section [3.1.5.45](#)). This parameter MUST be used by all following SQL queries in the batch, as the `@EidHome` parameter for **proc\_NavStructMoveNode** (section [3.1.5.50](#)) and **proc\_PutWebNavStructNode** (section [3.1.5.53](#)) and as the `@@EidHome` parameter for **proc\_NavStructAddNewNodeByDocId** (section [3.1.5.44](#)) and **proc\_NavStructAddNewNodeByUrl** (section [3.1.5.45](#)).

A part of the navigation structure of a site (2) can be inherited by its subsites. This part is referred to as the **inheritable navigation structure**. Particular parts of the navigation structure and the inheritable navigation structure can be cached for faster access.

### 3.1.1.6 TreeView

Data in the site (2) is stored in a hierarchical manner. A site (2) can contain subsites, lists (1), and document libraries. Lists (1) and document libraries can contain folders. A folder can contain other folders inside it. The Tree View tries to present a graphical representation of this structure. The protocol enables the front-end Web server to retrieve data according to the hierarchy.

### 3.1.1.7 Web Discussions

The protocol enables the user to comment and discuss documents stored on the server. The comments are stored in the back-end database server. They are not part of the document. Each of the comments has an identifier and is associated with a document. Other information stored in the server include the author of the comment, the date of last modification, and a client specified bookmark that enables the client to place the comment in the right position in the document. In the case of threaded discussions, comments have a parent comment associated with them. The parent comment is identified through its identifier that is stored in the reply.

### 3.1.2 Timers

An execution timeout timer on the protocol server governs the execution time for any requests. The amount of time is specified by a timeout value that is configured on the protocol server for all connections.

### 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

The T-SQL syntax for each stored procedure and result set, and the variables they are composed of, is defined in the [\[MSDN-TSQL-Ref\]](#) protocol. In the T-SQL syntax, the variable name is followed by the type of the variable that can optionally have a length value in brackets and can optionally have a default value indicated by an equals sign followed by the default value. Unless otherwise specified, all stored procedures defined in this section are located in the content database.

For definitional clarity, a name has been assigned to any **column (1)** in the result sets that do not have a defined name in their current implementation. This does not affect the operation of the result set, as the ordinal position of any column (1) with no defined name is expected by the front-end Web server. Such names are designated in the text using curly braces in the form (1) {*name*}.

#### 3.1.5.1 proc\_AddDocComment

The **proc\_AddDocComment** stored procedure is called to add a Web discussion comment to a document. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_AddDocComment (
    @SiteId          uniqueidentifier,
    @DocFullUrl      nvarchar(260),
    @ParentId        int,
    @CommentId       nvarchar(255),
```



```

    @Bookmark          nvarchar(127),
    @Author            nvarchar(255),
    @Subject           nvarchar(255),
    @Status            smallint,
    @Comment           nvarchar(max),
    @Size              int,
    @UserId            int,
    @UserTitle         nvarchar(255),
    @Created           datetime,
    @RequestGuid       uniqueidentifier = NULL OUTPUT
);

```

**@SiteId:** The site collection identifier for the site collection containing the specified document.

**@DocFullUrl:** The **store-relative URL** to the document.

**@ParentId:** The identifier of the Web discussion comment being replied to. If this Web discussion comment is not a reply, this SHOULD be zero ("0")[<3>](#).

**@CommentId:** A protocol client specified identifier for the Web discussion comment.

**@Bookmark:** A protocol client specified reference to the place in the document to which this Web discussion comment refers.

**@Author:** A protocol client specified name for the user that is adding this Web discussion comment.

**@Subject:** The subject of this Web discussion comment. This parameter MUST NOT be NULL.

**@Status:** A Web Discussion Comment Status Flag (section [2.2.2.3](#)) that indicates the status of a Web discussion comment.

**@Comment:** The body text of the Web discussion comment. This parameter MUST NOT be NULL.

**@Size:** The size in bytes of this Web discussion comment that is available for use in quota management. This parameter MUST NOT be NULL.

**@UserId:** A **user identifier** for the user adding the Web discussion comment.

**@UserTitle:** The display name of the user that is adding this Web discussion comment.

**@Created:** The time in **UTC** when this Web discussion comment was created. If this is NULL, the database server MUST use the current time in UTC.

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

**Return Values:** Returns an **integer return code** that MUST be listed in the following table:

Value	Description
"0"	Successful completion.
"3"	The document was not found in the specified site collection or <i>@SiteId</i> was NULL.
"212"	The specified site collection has been <b>locked</b> and writes are disallowed.
"1816"	The quota for the specified site collection has been exceeded.

**Result Sets:** MUST return zero or one result sets as defined in the following subsection.

### 3.1.5.1.1 AddDocComment Result Set

The **AddDocComment** result set returns information about the Web discussion comment that was added. If the return code is zero ("0"), the **AddDocComment** result set MUST be returned and MUST contain one row (3). If the return code is nonzero, the protocol server SHOULD NOT return this result set, but if returned, the protocol client MUST ignore it<4>. The T-SQL syntax for the result set is as follows:

```
{Created}          datetime,  
{Id}              int;
```

**{Created}:** The time in UTC when this Web discussion comment was added.

**{Id}:** The identifier of the newly added Web discussion comment.

### 3.1.5.2 proc\_AddSubscription

The **proc\_AddSubscription** stored procedure is called to add an alert (1). A list (1) **level** alert (1) is an alert (1) that is triggered for changes to any list item in a list (1) or when new list items are added to the list (1). An item level alert (1) is an alert (1) that is triggered for changes to a single list item for which an alert (1) was created. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_AddSubscription(  
    @SiteId                uniqueidentifier,  
    @NotifyFreq            int,  
    @NotifyTime            datetime,  
    @NotifyTimeUTC        datetime,  
    @Status                tinyint,  
    @WebId                 uniqueidentifier,  
    @ListId                uniqueidentifier,  
    @ItemId                int,  
    @EventType             int,  
    @UserId                int,  
    @SiteUrl               nvarchar(136),  
    @WebUrl                nvarchar(256),  
    @WebTitle              nvarchar(255),  
    @WebLanguage           int,  
    @WebLocale             int,  
    @WebTimeZone          smallint,  
    @WebTime24             bit,  
    @WebCalendarType      smallint,  
    @WebAdjustHijriDays   smallint,  
    @ListUrl               nvarchar(256),  
    @ListTitle             nvarchar(255),  
    @ListBaseType         int,  
    @ListServerTemplate   int,  
    @AlertTitle           nvarchar(1000),  
    @AlertType            int,  
    @AlertTemplateName    nvarchar(256),  
    @Filter                nvarchar(4000),  
    @BinaryFilter          varbinary(1024),  
    @Properties            nvarchar(max),  
    @ItemDocUrl           nvarchar(260),  
    @ItemDocId            uniqueidentifier,
```

```

    @DeliveryChannel      int,
    @Max                  int,
    @NewSubId             uniqueidentifier = NULL OUTPUT,
    @UserEmail            nvarchar(255) = NULL OUTPUT,
    @ItemName             nvarchar(255) = NULL OUTPUT,
    @RequestGuid          uniqueidentifier = NULL OUTPUT
);

```

**@SiteId:** The site collection identifier for the site collection in which the list (1) or item exists. *@SiteId* MUST be a GUID and MUST NOT be NULL.

**@NotifyFreq:** A notification frequency type (section [2.2.1.6](#)) that specifies an **integer** indicating the frequency of the alert (1) notification.

**@NotifyTime:** The time, in local time for the site (2) in which the list (1) or item exists, at which the alert (1) is supposed to fire. This MUST NOT be NULL.

**@NotifyTimeUTC:** The time in UTC at which the alert (1) is supposed to fire. This MUST NOT be NULL.

**@Status:** An alert (1) status type (section [2.2.1.2](#)) that specifies an **integer** indicating the status of the alert subscription.

**@WebId:** The site identifier for the site (2) in which the list (1) or item exist. This MUST NOT be NULL.

**@ListId:** The list identifier for a list (1) level alert (1) or the list (1) containing the list item for an item level alert (1). This MUST NOT be NULL.

**@ItemId:** The identifier of the list item for an item level alert (1). This MUST NOT be NULL if the alert (1) type (section [2.2.1.3](#)) of the alert subscription is an item type and MUST be NULL if the alert (1) type of the alert subscription is a list (1) type.

**@EventType:** An alert (1) event (2) type (section [2.2.2.1](#)) that specifies an **integer** mask for the types of events (2) on which to fire the notification.

**@UserId:** The user identifier of the user for whom the alert (1) is created. This MUST NOT be NULL.

**@SiteUrl:** The URL of the site collection in which the list (1) exists.

**@WebUrl:** The URL of the site (2) in which the list (1) exists.

**@WebTitle:** The title of the site (2) in which the list (1) or item exists.

**@WebLanguage:** The **language code identifier (LCID)** of the display language of the site (2) containing the list (1). This MUST NOT be NULL.

**@WebLocale:** An **integer** representing the LCID of the site (2) **locale**. This MUST NOT be NULL.

**@WebTimeZone:** The time zone identifier for the time zone to be used when displaying time values for this site (2). This MUST NOT be NULL.

**@WebTime24:** A Boolean indicating if the Time is in a **24-hour clock notation** format or a **12-hour clock notation** format. This MUST NOT be NULL.

**@WebCalendarType:** The **calendar type** for the site (2). This MUST NOT be NULL.

**@WebAdjustHijriDays:** The number of days to extend or reduce the current month in Hijri calendars on the site (2) with which this alert subscription is associated. This value MUST NOT be NULL.

**@ListUrl:** The URL of the list (1).

**@ListTitle:** The title of the list (1).

**@ListBaseType:** The base type of the list (1) with which this alert subscription is associated. See [\[MS-WSSFO2\]](#) section 2.2.3.11.

**@ListServerTemplate:** The **list template** of the list (1). See [\[MS-WSSFO2\]](#) section 2.2.3.12 for details.

**@AlertTitle:** The title of the alert (1) for display in UI.

**@AlertType:** An **integer** whose lower 8 bits specifies the type of the alert (1), as specified in section [2.2.1.3](#). This MUST also contain the special alert flags (section [2.2.2.2](#)) for an always notify alert or a system alert. This MUST NOT be NULL.

**@AlertTemplateName:** The name of the **alert template** that is used for formatting the notification.

**@Filter:** The XML query filter used to further filter out results from the matching events (2) for the list (1) and list item.

**@BinaryFilter:** An implementation-specific binary format of the filter specified by the *@Filter* parameter.

**@Properties:** An XML blob representing the properties of the alert (1).

**@ItemDocUrl:** The store-relative URL of the list item for which the alert (1) is being added.

**@ItemDocId:** The identifier of the list item in the **Docs** view.

**@DeliveryChannel:** Specifies the delivery channel for the alert subscription as defined by section [2.2.1.1](#).

**@Max:** The maximum number of alerts (1) allowed in a content database.

**@NewSubId:** The GUID of the alert (1) that was added returned as an output parameter.

**@UserEmail:** The e-mail address of the user who receives notifications returned as an output parameter.

**@ItemName:** The name of the item for item level alerts (1) returned as an output parameter.

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

**Return Values:** Returns an **integer** return code that MUST be in the following table:

Value	Description
"0"	Successful completion.
"31"	General Error
"68"	Subscription quota is being exceeded

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

### 3.1.5.3 proc\_AddUserResource

The **proc\_AddUserResource** stored procedure is called to add a user resource string. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_AddUserResource (  
    @WebId                uniqueidentifier,  
    @ListId               uniqueidentifier,  
    @ResourceName         nvarchar(520),  
    @ResourceCategory     int,  
    @BitType              bit,  
    @LCID                 int,  
    @NVarCharVal          nvarchar(256),  
    @NtextVal             nvarchar(max)  
);
```

**@WebId:** The identifier of the site (2) that contains the user resource string. It MUST NOT be NULL.

**@ListId:** The identifier of the list (1) that contains the user resource string. If the resource string does not belong to a list (1), it MUST be an **empty GUID**. It MUST NOT be NULL.

**@ResourceName:** The name of the user resource string. It MUST NOT be NULL.

**@ResourceCategory:** The category of the user resource string. It MUST be one of the following values:

Value	Description	Constraint on the @ResourceName
"0"	The resource string is defined by the end user.	The value of @ResourceName MUST begin with a digit or alphabetic letter.
"1"	The resource string is for the title of the site (2) specified by @WebId.	The value of @ResourceName MUST be "_WebTitle".
"2"	The resource string is for the description of the site (2) specified by @WebId.	The value of @ResourceName MUST be "_WebDescription".
"3"	The resource string is for the title of the list (1) specified by @ListId.	The value of @ResourceName MUST be "_ListTitle".
"4"	The resource string is for the description of the list (1) specified by @ListId.	The value of @ResourceName MUST be "_ListDescription".
"5"	The resource string is for the title of a navigation node.	The value of @ResourceName MUST begin with the <b>string</b> "_NavNode" and then be followed by the string representation of the navigation node element identifier. For example, if the navigation node element identifier is "1002", its value is "_NavNode1002".

**@BitType:** Type of the resource as defined in user resource type (section [2.2.1.10](#)). It MUST NOT be NULL.

**@LCID:** The LCID for the user resource string. It MUST NOT be NULL.

**@NVarCharVal:** The value of the user resource string to be added when *@BitType* is zero ("0"). It MUST be NULL when *@BitType* is "1".

**@NtextVal:** The value of the user resource string to be added when *@BitType* is "1". It MUST be NULL when *@BitType* is zero ("0").

**Return values:** Returns an **integer** that MUST be zero.

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

### 3.1.5.4 **proc\_CheckMeetingInstance**

The **proc\_CheckMeetingInstance** stored procedure is called to determine whether a given meeting instance exists in the Meeting Workspace site and is instantiated. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_CheckMeetingInstance(  
    @WebId                uniqueidentifier,  
    @ReturnSeriesItem    bit,  
    @InstanceID          int,  
    @IsInstantiated      bit OUTPUT,  
    @DTStartUTC          datetime OUTPUT,  
    @RequestGuid         uniqueidentifier = NULL OUTPUT  
);
```

**@WebId:** The identifier of an instance of a Meeting Workspace site. *@WebId* MUST NOT be NULL.

**@ReturnSeriesItem:** Specifies whether or not to return the meeting series item as a result set. Zero ("0") means that no result set is returned.

**@InstanceID:** Specifies the identifier of the meeting instance to be checked. This MUST NOT be NULL.

**@IsInstantiated:** Specifies if the meeting is instantiated. Zero ("0") means that the meeting is not instantiated; "1" means that the meeting is instantiated. This value is filled and returned to the caller.

**@DTStartUTC:** Specifies the beginning date of the meeting, in UTC. This value is filled and returned to the caller.

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

**Return Values:** Returns an **integer** that MUST be in the following table:

Value	Description
"0"	Successful completion. The meeting instance exists and is instantiated.
"2"	Three possible failure conditions: <ul style="list-style-type: none"><li>Meeting instance found, but not instantiated.</li><li>Meeting instance is a valid instance of the recurring meeting but not instantiated and <b>ReturnSeriesItem</b> equals zero ("0").</li><li>No valid meeting calendar event data found.</li></ul>

Value	Description
"13"	Unable to retrieve the meeting list (1). Output parameters <i>@IsInstantiated</i> and <i>@DTStartUTC</i> are NULL.

**Result Sets:** MUST return a single result set on success when all of the following are true:

- The *ReturnSeriesItem* parameter is set to "1".
- The meeting instance is a recurring calendar event.
- The meeting instance is not already instantiated.

### 3.1.5.4.1 Meeting Event Result Set

**proc\_CheckMeetingInstance** returns a result set that MUST contain one item that is the meeting item that represents a series.

```

bit3                bit,
tp_InstanceID      int,
int1                int,
uniqueidentifier1  uniqueidentifier,
datetime3           datetime,
datetime1           datetime,
datetime2           datetime,
int2                int,
ntext1              ntext,
ntext2              ntext,
ntext3              ntext,
datetime5           datetime,
int3                int;

```

For more details on the columns (1) in **UserData** view, see [\[MS-WSSFO2\]](#) section 2.2.7.8.

**bit3:** Specifies whether this is a standalone or recurring meeting. Zero ("0") means a standalone meeting, "1" means a recurring meeting. Because this stored procedure is for a recurring meeting, this field (3) MUST be "1" for the row (3) returned.

**tp\_InstanceID:** The identifier for this meeting instance. For an instance of a recurring meeting this is the beginning date of the meeting in UTC. For the default instance of a recurring meeting this field (3) is zero ("0"). Because the result set returned is always for the default instance of a recurring meeting associated with the specified Meeting Workspace site, this MUST return a zero.

**int1:** Specifies the calendar event type for this meeting as defined by meetings calendar event type (section [2.2.1.4](#)). By definition this field (3) is "1" in the row (3) returned, indicating a recurring calendar event.

**uniqueidentifier1:** Specifies the unique identifier for this meeting series.

**datetime3:** Specifies the recurrence identifier of the meeting instance. For a recurring meeting instance this is the beginning date of the meeting series in UTC. For a meeting instance exception, this is the original starting **datetime** of the meeting instance in UTC. For the default meeting instance or standalone meetings, this field (3) is NULL. Because the result set returned by this stored procedure is always for the default instance of a recurring meeting associated with specified Meeting Workspace site, this field (3) MUST return a NULL.

**datetime1:** The start date for this meeting instance in UTC.

**datetime2:** The end date for this meeting instance in UTC. For the default instance of a recurring meeting associated with specified Meeting Workspace site that is returned by this stored procedure, this field (3) is set to the maximum SQL datetime.

**int2:** Specifies the duration of a meeting instance, in the recurring meeting series, in seconds.

**ntext1:** For the default meeting instance, this is the **XML fragment** describing the **recurrence rule** for the recurring meeting series.

**ntext2:** This returns any recurrence rule specified for the recurring meeting.

**ntext3:** This property returns a rule or repeating pattern defined for an exception to the specified recurring meeting.

**datetime5:** If the series has been suppressed, this is the date it has been suppressed until. When this field (3) is NULL it indicates that the series has not been suppressed.

**int3:** The time zone identifier used when creating this meeting.

### 3.1.5.5 proc\_CheckNavStructContainsPage

The **proc\_CheckNavStructContainsPage** stored procedure is called to determine if a document in a site (2) exists in the navigation structure of the site (2). The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_CheckNavStructContainsPage (  
    @SiteId                uniqueidentifier,  
    @DirName               nvarchar(256),  
    @LeafName              nvarchar(128),  
    @RequestGuid           uniqueidentifier = NULL OUTPUT  
);
```

**@SiteId:** The site collection identifier for a site collection that contains the specified document.

**@DirName:** The directory name of the requested document in **store-relative form**. If the requested document is in the root directory, the value **MUST** be an **empty string (1)**.

**@LeafName:** The leaf name of the requested document in store-relative form.

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

**Return Values:** Returns an **integer** return code that **MUST** be in the following table:

Value	Description
"0"	The document does not exist in the navigation structure.
"1"	The document exists in the navigation structure.

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



### 3.1.5.6 proc\_DeleteSubscription

The **proc\_DeleteSubscription** stored procedure is called to delete an alert (1). The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_DeleteSubscription(  
    @SiteId          uniqueidentifier,  
    @SubId           uniqueidentifier,  
    @UserId          int,  
    @RequestGuid    uniqueidentifier = NULL OUTPUT  
);
```

**@SiteId:** Specifies the site collection identifier for the site collection. This MUST NOT be NULL.

**@SubId:** Specifies the identifier of the alert subscription. This MUST NOT be NULL.

**@UserId:** The user identifier for the user for whom the alert (1) has been created.

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

**Return Values:** Returns an **integer** return code that MUST be listed in the following table:

Value	Description
"0"	Successful completion.
"31"	General Error

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

### 3.1.5.7 proc\_DeleteSubscriptionJunctionEntries

The **proc\_DeleteSubscriptionJunctionEntries** stored procedure is called to clean up junction entries for up to 256 alert subscription identifiers after the relevant alerts (1), either immediate or scheduled, have been reported via e-mail. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_DeleteSubscriptionJunctionEntries(  
    @f001          uniqueidentifier = NULL,  
    @f002          uniqueidentifier = NULL,  
    @f003          uniqueidentifier = NULL,  
    @f004          uniqueidentifier = NULL,  
    @f005          uniqueidentifier = NULL,  
    @f006          uniqueidentifier = NULL,  
    @f007          uniqueidentifier = NULL,  
    @f008          uniqueidentifier = NULL,  
    @f009          uniqueidentifier = NULL,  
    @f010          uniqueidentifier = NULL,  
    @f011          uniqueidentifier = NULL,  
    @f012          uniqueidentifier = NULL,  
    @f013          uniqueidentifier = NULL,  
    @f014          uniqueidentifier = NULL,  
    @f015          uniqueidentifier = NULL,  
    @f016          uniqueidentifier = NULL,  
    @f017          uniqueidentifier = NULL,  
    @f018          uniqueidentifier = NULL,  
    @f019          uniqueidentifier = NULL,
```

@f020           uniqueidentifier = NULL,  
@f021           uniqueidentifier = NULL,  
@f022           uniqueidentifier = NULL,  
@f023           uniqueidentifier = NULL,  
@f024           uniqueidentifier = NULL,  
@f025           uniqueidentifier = NULL,  
@f026           uniqueidentifier = NULL,  
@f027           uniqueidentifier = NULL,  
@f028           uniqueidentifier = NULL,  
@f029           uniqueidentifier = NULL,  
@f030           uniqueidentifier = NULL,  
@f031           uniqueidentifier = NULL,  
@f032           uniqueidentifier = NULL,  
@f033           uniqueidentifier = NULL,  
@f034           uniqueidentifier = NULL,  
@f035           uniqueidentifier = NULL,  
@f036           uniqueidentifier = NULL,  
@f037           uniqueidentifier = NULL,  
@f038           uniqueidentifier = NULL,  
@f039           uniqueidentifier = NULL,  
@f040           uniqueidentifier = NULL,  
@f041           uniqueidentifier = NULL,  
@f042           uniqueidentifier = NULL,  
@f043           uniqueidentifier = NULL,  
@f044           uniqueidentifier = NULL,  
@f045           uniqueidentifier = NULL,  
@f046           uniqueidentifier = NULL,  
@f047           uniqueidentifier = NULL,  
@f048           uniqueidentifier = NULL,  
@f049           uniqueidentifier = NULL,  
@f050           uniqueidentifier = NULL,  
@f051           uniqueidentifier = NULL,  
@f052           uniqueidentifier = NULL,  
@f053           uniqueidentifier = NULL,  
@f054           uniqueidentifier = NULL,  
@f055           uniqueidentifier = NULL,  
@f056           uniqueidentifier = NULL,  
@f057           uniqueidentifier = NULL,  
@f058           uniqueidentifier = NULL,  
@f059           uniqueidentifier = NULL,  
@f060           uniqueidentifier = NULL,  
@f061           uniqueidentifier = NULL,  
@f062           uniqueidentifier = NULL,  
@f063           uniqueidentifier = NULL,  
@f064           uniqueidentifier = NULL,  
@f065           uniqueidentifier = NULL,  
@f066           uniqueidentifier = NULL,  
@f067           uniqueidentifier = NULL,  
@f068           uniqueidentifier = NULL,  
@f069           uniqueidentifier = NULL,  
@f070           uniqueidentifier = NULL,  
@f071           uniqueidentifier = NULL,  
@f072           uniqueidentifier = NULL,  
@f073           uniqueidentifier = NULL,  
@f074           uniqueidentifier = NULL,  
@f075           uniqueidentifier = NULL,  
@f076           uniqueidentifier = NULL,  
@f077           uniqueidentifier = NULL,  
@f078           uniqueidentifier = NULL,

@f079            uniqueidentifier = NULL,  
@f080            uniqueidentifier = NULL,  
@f081            uniqueidentifier = NULL,  
@f082            uniqueidentifier = NULL,  
@f083            uniqueidentifier = NULL,  
@f084            uniqueidentifier = NULL,  
@f085            uniqueidentifier = NULL,  
@f086            uniqueidentifier = NULL,  
@f087            uniqueidentifier = NULL,  
@f088            uniqueidentifier = NULL,  
@f089            uniqueidentifier = NULL,  
@f090            uniqueidentifier = NULL,  
@f091            uniqueidentifier = NULL,  
@f092            uniqueidentifier = NULL,  
@f093            uniqueidentifier = NULL,  
@f094            uniqueidentifier = NULL,  
@f095            uniqueidentifier = NULL,  
@f096            uniqueidentifier = NULL,  
@f097            uniqueidentifier = NULL,  
@f098            uniqueidentifier = NULL,  
@f099            uniqueidentifier = NULL,  
@f100            uniqueidentifier = NULL,  
@f101            uniqueidentifier = NULL,  
@f102            uniqueidentifier = NULL,  
@f103            uniqueidentifier = NULL,  
@f104            uniqueidentifier = NULL,  
@f105            uniqueidentifier = NULL,  
@f106            uniqueidentifier = NULL,  
@f107            uniqueidentifier = NULL,  
@f108            uniqueidentifier = NULL,  
@f109            uniqueidentifier = NULL,  
@f110            uniqueidentifier = NULL,  
@f111            uniqueidentifier = NULL,  
@f112            uniqueidentifier = NULL,  
@f113            uniqueidentifier = NULL,  
@f114            uniqueidentifier = NULL,  
@f115            uniqueidentifier = NULL,  
@f116            uniqueidentifier = NULL,  
@f117            uniqueidentifier = NULL,  
@f118            uniqueidentifier = NULL,  
@f119            uniqueidentifier = NULL,  
@f120            uniqueidentifier = NULL,  
@f121            uniqueidentifier = NULL,  
@f122            uniqueidentifier = NULL,  
@f123            uniqueidentifier = NULL,  
@f124            uniqueidentifier = NULL,  
@f125            uniqueidentifier = NULL,  
@f126            uniqueidentifier = NULL,  
@f127            uniqueidentifier = NULL,  
@f128            uniqueidentifier = NULL,  
@f129            uniqueidentifier = NULL,  
@f130            uniqueidentifier = NULL,  
@f131            uniqueidentifier = NULL,  
@f132            uniqueidentifier = NULL,  
@f133            uniqueidentifier = NULL,  
@f134            uniqueidentifier = NULL,  
@f135            uniqueidentifier = NULL,  
@f136            uniqueidentifier = NULL,  
@f137            uniqueidentifier = NULL,

@f138           uniqueidentifier = NULL,  
@f139           uniqueidentifier = NULL,  
@f140           uniqueidentifier = NULL,  
@f141           uniqueidentifier = NULL,  
@f142           uniqueidentifier = NULL,  
@f143           uniqueidentifier = NULL,  
@f144           uniqueidentifier = NULL,  
@f145           uniqueidentifier = NULL,  
@f146           uniqueidentifier = NULL,  
@f147           uniqueidentifier = NULL,  
@f148           uniqueidentifier = NULL,  
@f149           uniqueidentifier = NULL,  
@f150           uniqueidentifier = NULL,  
@f151           uniqueidentifier = NULL,  
@f152           uniqueidentifier = NULL,  
@f153           uniqueidentifier = NULL,  
@f154           uniqueidentifier = NULL,  
@f155           uniqueidentifier = NULL,  
@f156           uniqueidentifier = NULL,  
@f157           uniqueidentifier = NULL,  
@f158           uniqueidentifier = NULL,  
@f159           uniqueidentifier = NULL,  
@f160           uniqueidentifier = NULL,  
@f161           uniqueidentifier = NULL,  
@f162           uniqueidentifier = NULL,  
@f163           uniqueidentifier = NULL,  
@f164           uniqueidentifier = NULL,  
@f165           uniqueidentifier = NULL,  
@f166           uniqueidentifier = NULL,  
@f167           uniqueidentifier = NULL,  
@f168           uniqueidentifier = NULL,  
@f169           uniqueidentifier = NULL,  
@f170           uniqueidentifier = NULL,  
@f171           uniqueidentifier = NULL,  
@f172           uniqueidentifier = NULL,  
@f173           uniqueidentifier = NULL,  
@f174           uniqueidentifier = NULL,  
@f175           uniqueidentifier = NULL,  
@f176           uniqueidentifier = NULL,  
@f177           uniqueidentifier = NULL,  
@f178           uniqueidentifier = NULL,  
@f179           uniqueidentifier = NULL,  
@f180           uniqueidentifier = NULL,  
@f181           uniqueidentifier = NULL,  
@f182           uniqueidentifier = NULL,  
@f183           uniqueidentifier = NULL,  
@f184           uniqueidentifier = NULL,  
@f185           uniqueidentifier = NULL,  
@f186           uniqueidentifier = NULL,  
@f187           uniqueidentifier = NULL,  
@f188           uniqueidentifier = NULL,  
@f189           uniqueidentifier = NULL,  
@f190           uniqueidentifier = NULL,  
@f191           uniqueidentifier = NULL,  
@f192           uniqueidentifier = NULL,  
@f193           uniqueidentifier = NULL,  
@f194           uniqueidentifier = NULL,  
@f195           uniqueidentifier = NULL,  
@f196           uniqueidentifier = NULL,

@f197            uniqueidentifier = NULL,  
@f198            uniqueidentifier = NULL,  
@f199            uniqueidentifier = NULL,  
@f200            uniqueidentifier = NULL,  
@f201            uniqueidentifier = NULL,  
@f202            uniqueidentifier = NULL,  
@f203            uniqueidentifier = NULL,  
@f204            uniqueidentifier = NULL,  
@f205            uniqueidentifier = NULL,  
@f206            uniqueidentifier = NULL,  
@f207            uniqueidentifier = NULL,  
@f208            uniqueidentifier = NULL,  
@f209            uniqueidentifier = NULL,  
@f210            uniqueidentifier = NULL,  
@f211            uniqueidentifier = NULL,  
@f212            uniqueidentifier = NULL,  
@f213            uniqueidentifier = NULL,  
@f214            uniqueidentifier = NULL,  
@f215            uniqueidentifier = NULL,  
@f216            uniqueidentifier = NULL,  
@f217            uniqueidentifier = NULL,  
@f218            uniqueidentifier = NULL,  
@f219            uniqueidentifier = NULL,  
@f220            uniqueidentifier = NULL,  
@f221            uniqueidentifier = NULL,  
@f222            uniqueidentifier = NULL,  
@f223            uniqueidentifier = NULL,  
@f224            uniqueidentifier = NULL,  
@f225            uniqueidentifier = NULL,  
@f226            uniqueidentifier = NULL,  
@f227            uniqueidentifier = NULL,  
@f228            uniqueidentifier = NULL,  
@f229            uniqueidentifier = NULL,  
@f230            uniqueidentifier = NULL,  
@f231            uniqueidentifier = NULL,  
@f232            uniqueidentifier = NULL,  
@f233            uniqueidentifier = NULL,  
@f234            uniqueidentifier = NULL,  
@f235            uniqueidentifier = NULL,  
@f236            uniqueidentifier = NULL,  
@f237            uniqueidentifier = NULL,  
@f238            uniqueidentifier = NULL,  
@f239            uniqueidentifier = NULL,  
@f240            uniqueidentifier = NULL,  
@f241            uniqueidentifier = NULL,  
@f242            uniqueidentifier = NULL,  
@f243            uniqueidentifier = NULL,  
@f244            uniqueidentifier = NULL,  
@f245            uniqueidentifier = NULL,  
@f246            uniqueidentifier = NULL,  
@f247            uniqueidentifier = NULL,  
@f248            uniqueidentifier = NULL,  
@f249            uniqueidentifier = NULL,  
@f250            uniqueidentifier = NULL,  
@f251            uniqueidentifier = NULL,  
@f252            uniqueidentifier = NULL,  
@f253            uniqueidentifier = NULL,  
@f254            uniqueidentifier = NULL,  
@f255            uniqueidentifier = NULL,

```

        @f256                uniqueidentifier = NULL,
        @RequestGuid        uniqueidentifier = NULL OUTPUT
    );

```

**@fnnn:** The alert subscription identifier for which the alert subscription data is to be deleted.

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

**Return Values:** The stored procedure MUST return an **integer** return code that MUST be zero.

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

### 3.1.5.8 proc\_EnumDoclibsFileDlg

The **proc\_EnumDoclibsFileDlg** stored procedure is called to enumerate certain properties of document library items in a given site (2) of the specified site collection. These properties are described in sections [3.1.5.8.1](#) and [3.1.5.8.2](#).

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

```

PROCEDURE proc_EnumDoclibsFileDlg(
    @SiteId                uniqueidentifier,
    @WebId                 uniqueidentifier,
    @Collation              nvarchar(32),
    @GetTemplate           bit,
    @OnlyLibsWithTemplates bit,
    @IncludeSubWebs       bit,
    @Scopes                varbinary(max) = NULL,
    @RequestGuid           uniqueidentifier = NULL OUTPUT
);

```

**@SiteId:** The GUID of the site collection.

**@WebId:** The identifier of the site (2) containing the requested document libraries.

**@Collation:** A **Windows collation name** string identifier that follows the format for the T-SQL COLLATE clause. This MUST be the **collation** name of one of the valid **collation order** values, with the **case-insensitive** and **accent-sensitive** flags set. For example, the default collation order is "25", Latin1\_General, which has a **SQL collation name** string of "Latin1\_General\_CI\_AS" with the **case-insensitive** and **accent-sensitive** flags set.

The results in the result set are ordered on the **tp\_Title** column (1) using this collation.

**@GetTemplate:** A bit flag that specifies whether the result set should include **document templates** associated with the library. A value of "1" specifies that **Templates** MUST be included.

**@OnlyLibsWithTemplates:** A bit flag (zero ("0") or "1") that specifies whether the result set should include only document libraries that have an associated template. A value of "1" specifies the result set MUST include only libraries that have an associated template.

**@IncludeSubWebs:** A bit flag (zero ("0") or "1") that indicates whether the result set should return properties of document libraries or properties of document libraries within subsites of the site (2) specified by *@WebId*. A value of "1" specifies that the result set MUST include only properties of subsites. A value of zero specifies that the result set MUST include only properties of document libraries.

**@Scopes:** A binary blob containing a concatenation of one or more GUIDs, represented as 16-byte binary strings with no delimiters. These 16-byte fragments **MUST** be convertible to **uniqueidentifiers**. There can be any number of 16-byte fragments in the image. If the value is NULL or this parameter is not specified, all scopes are included. The results in the result set are restricted to document libraries and subsites matching these GUID values.

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

**Return Values:** Returns an **integer** that **MUST** be zero.

**Result Sets:** Returns one of the result sets listed in the following table.

Input Parameter	Returned Result Set
@ GetTemplate = "1"	One <b>EnumerationWithTemplates</b> result set
@ GetTemplate = "0"	One <b>EnumerationWithoutTemplates</b> result set

### 3.1.5.8.1 EnumerationWithTemplates Result Set

The **EnumerationWithTemplates** result set contains certain properties of document libraries in a site (2).

This result set **MUST** contain one or more rows (3) if there are matches for the specified parameters, and **MUST** contain zero rows (3) if there were no matches for the specified parameters.

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

```
tp_RootFolder          nvarchar(386),
Docs#Template#FullUrl  nvarchar(386),
tp_Title               nvarchar(255),
tp_Description         ntext,
tp_ImageUrl            nvarchar(255);
```

**tp\_RootFolder:** The store-relative URL of the **root folder** of the document library. This value **MUST NOT** be NULL.

**Docs#Template#FullUrl:** The full store-relative URL for the Template if applicable. If **@IncludeSubWebs** is "1", this value **MUST** be NULL.

**tp\_Title:** The title of the document library. This value **MUST NOT** be NULL.

**tp\_Description:** The description of the document library. This value **MUST NOT** be NULL.

**tp\_ImageUrl:** The store-relative URL of an image icon used to represent the document library. This value **MUST NOT** be NULL.

### 3.1.5.8.2 EnumerationWithoutTemplates Result Set

The **EnumerationWithoutTemplates** result set contains certain properties of document libraries in a site (2).

This result set **MUST** contain one or more rows (3) if there are matches for the specified parameters, and **MUST** contain zero rows (3) if there were no matches for the specified parameters.

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

tp_RootFolder	nvarchar(386),
tp_Title	nvarchar(255),
tp_Description	ntext,
tp_ImageUrl	nvarchar(255);

**tp\_RootFolder:** The store-relative URL of the root folder of the document library. This value MUST not be NULL.

**tp\_Title:** The title of the document library. This value MUST NOT be NULL.

**tp\_Description:** The description of the document library. This value MUST NOT be NULL.

**tp\_ImageUrl:** The store-relative URL of an image icon used to represent the document library. This value MUST NOT be NULL.

### 3.1.5.9 proc\_EnumEmailAliases

The **proc\_EnumEmailAliases** stored procedure is invoked to retrieve all of the e-mail aliases in the content database. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_EnumEmailAliases();
```

**Return Values:** Returns an **integer** return code that MUST be zero.

**Result Sets:** MUST return a single EmailAliases result set as defined in section [2.2.4.1](#).

### 3.1.5.10 proc\_EnumEmailAliasesBySite

The **proc\_EnumEmailAliasesBySite** stored procedure is called to retrieve all of the e-mail aliases from the specified site collection. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_EnumEmailAliasesBySite(
    @SiteId          uniqueidentifier
);
```

**@SiteId:** The GUID of the site collection

**Return Values:** Returns an **integer** return code that MUST be zero.

**Result Sets:** MUST return a single EmailAliases result set as specified in section [2.2.4.1](#).

### 3.1.5.11 proc\_EnumResourcesAtScope

The **proc\_EnumResourcesAtScope** stored procedure is invoked to retrieve the resources for a given locale for a list (1) or site (2). The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_EnumResourcesAtScope (
    @WebId          uniqueidentifier,
    @ListId         uniqueidentifier,
    @LCID           int
);
```

**@WebId:** The identifier of the site (2) containing the requested resources.



**@ListId:** The identifier of the list (1) containing the requested resources, if the resources are scoped to the list (1). If the resources are scoped to the site (2), **ListId** MUST be null.

**@LCID:** The LCID of the locale of the requested resources. This MUST NOT be NULL.

**Return values:** MUST return an **integer** return code that MUST be zero.

**Result Sets:** This MUST return an **EnumResAtScopeResultSet** result set.

### 3.1.5.11.1 EnumResAtScopeResultSet Result Set

The **EnumResAtScopeResultSet** result set returns a list (1) of resources for the given list (1) or site (2). The number of rows (3) is equal to the number of resources at the given list (1) or site (2). The **EnumResValueLangResultSet** result set is defined by T-SQL syntax as follows:

```
ResourceName          nvarchar(520) NOT NULL,  
BitType               bit NOT NULL,  
BitDirty              bit NOT NULL,  
{NvarcharVal}        nvarchar(256) NULL,  
{NtextVal}           nvarchar(max) NULL;
```

**ResourceName:** The name of the resource.

**BitType:** This MUST be zero ("0") if the resource is a single line or less, and MUST be "1" if the resource is more than a single line. If **BitType** is zero ("0"), **NtextVal** MUST be NULL. If **BitType** is "1", **NvarcharVal** MUST be NULL.

**BitDirty:** This value is "1" if the resource is **dirty**, zero ("0") otherwise.

**NvarcharVal:** The text of the resource if **BitType** is zero ("0"). Otherwise, it MUST be NULL. Either this or **NtextVal** MUST be NULL.

**NtextVal:** The text of the resource if **BitType** is "1". Otherwise, it MUST be NULL. Either this or **NvarcharVal** MUST be NULL.

### 3.1.5.12 proc\_EnumResourceValuesForAllLangs

Enumerates all localized versions of a specified resource. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_EnumResourceValuesForAllLangs (  
    @WebId                uniqueidentifier,  
    @ListId               uniqueidentifier,  
    @ResourceName         nvarchar(520)  
)  
;
```

**@WebId:** The identifier of the site (2) containing the requested resource.

**@ListId:** The identifier of the list (1) containing the resource if the resource is scoped to the list (1). If the resource is scoped to the site (2), **ListId** MUST be NULL.

**@ResourceName:** The name of the resource.

**Return values:** MUST return an **integer** return code that MUST be zero.

**Result Sets:** This MUST return an **EnumResValueLangResultSet** result set.

### 3.1.5.12.1 EnumResValueLangResultSet Result Set

The **EnumResValueLangResultSet** result set is defined by T-SQL syntax as follows:

```
BitDirty          bit NOT NULL,  
LCID              int NOT NULL,  
{NvarcharVal}    nvarchar(256) NULL,  
{NtextVal}       nvarchar(max) NULL;
```

**BitDirty:** This value is "1" if the resource is dirty, zero otherwise.

**LCID:** The LCID of the resource.

**NvarcharVal:** The text of the resource if it is one line or less in length. Otherwise, it MUST be NULL. Either this or NTextVal MUST be NULL.

**NtextVal:** The text of the resource if it is more than one line in length. Otherwise, it MUST be NULL. Either this or NvarcharVal MUST be NULL.

### 3.1.5.13 proc\_EnumSubscribedSites

The **proc\_EnumSubscribedSites** stored procedure is called to enumerate the sites (2) that have been subscribed for alerts (1). The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_EnumSubscribedSites (  
    @NotifyFreq      int,  
    @bAlwaysNotify   bit,  
    @RequestGuid     uniqueidentifier = NULL OUTPUT  
);
```

**@NotifyFreq:** A notification frequency type (section [2.2.1.6](#)) that specifies an **integer** indicating the frequency of the alert (1) notification.

**@bAlwaysNotify:** A bit that indicates whether the alert (1) is an always notify alert.

**@RequestGuid:** Optional request identifier for the current request.

**Return Values:** MUST return an **integer** return code that MUST be zero.

**Result Sets:** MUST return one result set as defined in the following subsection.

#### 3.1.5.13.1 SubscribedSites Result Set

The **SubscribedSites** result set returns the list (1) of site identifiers that have been subscribed. The **SubscribedSites** result set contains zero or more rows (3). The T-SQL syntax for the result set is as follows:

```
SiteId            uniqueidentifier;
```

**SiteId:** The site identifier for the site (2).

### 3.1.5.14 proc\_GetAlertsSqmData

The **proc\_GetAlertsSqmData** stored procedure is called to retrieve the number of alerts (1) in the content database. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetAlertsSqmData(  
    @RequestGuid          uniqueidentifier = NULL OUTPUT  
);
```

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

**Return Values:** Returns an **integer** return code. The protocol client MUST ignore the return code returned by the **proc\_GetAlertsSqmData** stored procedure.

**Result Sets:** MUST return a single row (3) result set.

#### 3.1.5.14.1 AlertsSqmData Result Set

The **AlertsSqmData** result set contains the number of alerts (1) in the content database. The **AlertsSqmData** result set MUST contain one row (3) of five columns (1). The **AlertsSqmData** result set is defined using T-SQL syntax, as follows:

```
NumImmedAlerts          int NOT NULL,  
NumSchedAlerts          int NOT NULL,  
NumAlertsItem           int NOT NULL,  
NumAlertsList           int NOT NULL,  
NumAlertsCustom         int NOT NULL;
```

**NumImmedAlerts:** The number of immediate alerts (1). This value MUST NOT be NULL.

**NumSchedAlerts:** The number of scheduled alerts (1). This value MUST NOT be NULL.

**NumAlertsItem:** The number of alerts (1) of type Item alert (section [2.2.1.3](#)). This value MUST NOT be NULL.

**NumAlertsList:** The number of alerts (1) of type List alert (section [2.2.1.3](#)). This value MUST NOT be NULL.

**NumAlertsCustom:** The number of alerts (1) of type Custom alert (section [2.2.1.3](#)). This value MUST NOT be NULL.

### 3.1.5.15 proc\_GetDefaultMtgInstance

The **proc\_GetDefaultMtgInstance** stored procedure is called to return a result set of all meeting instances tied to the supplied Meeting Workspace site. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetDefaultMtgInstance(  
    @WebId                uniqueidentifier,  
    @RequestGuid          uniqueidentifier = NULL OUTPUT  
);
```

**@WebId:** The identifier of an instance of a Meeting Workspace site.

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

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

Value	Description
"0"	Successful completion. Result set MUST have one or more <b>records</b> .
"13"	No meetings list (1) was found on the workspace specified by <i>@WebId</i> .

**Result Sets:** MUST return one result set on successful completion.

### 3.1.5.15.1 Default Meeting Instance Result Set

The DefaultMeetingInstance result set contains the properties of the meeting instance.

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

```
bit3                bit,  
tp_InstanceID      int,  
int1                int,  
uniqueidentifier1  uniqueidentifier,  
datetime3           datetime,  
datetime1           datetime,  
datetime2           datetime,  
int2                int,  
ntext1              ntext,  
ntext2              ntext,  
ntext3              ntext,  
datetime5           datetime,  
int3                int;
```

For more details on the columns (1) in **UserData** view, see [\[MS-WSSFO2\]](#) section 2.2.7.8.

**bit3:** Specifies whether this is a standalone or recurring meeting. Zero ("0") means a standalone meeting, "1" means a recurring meeting.

**tp\_InstanceID:** Specifies the identifier for this meeting instance in the Meeting Workspace site. For a recurring meeting instance this is the start date of the meeting in UTC. For the default meeting instance, this field (3) is zero. For standalone meetings, this field (3) is "1" for the first instance and is incremented by 1 for every additional standalone meeting associated with the same Meeting Workspace site.

**int1:** Specifies the calendar event type for this meeting as defined in section [2.2.1.4](#).

**uniqueidentifier1:** Specifies the unique identifier for this meeting series.

**datetime3:** Specifies the recurrence identifier of the meeting instance in UTC. For a recurring meeting instance, this is the start date of the meeting series. For a meeting instance exception, this is the original starting **datetime** of the meeting instance. For the default meeting instance or standalone meetings, this field (3) MUST be NULL.

**datetime1:** Specifies the start date for this meeting instance in UTC.

**datetime2:** The end datetime for this meeting instance in UTC. For the default meeting instance tied to a Meeting Workspace site, this is set to the maximum SQL datetime.

**int2:** Specifies the duration of a meeting instance, in seconds. For recurring meetings tied to Meeting Workspace sites, only the default instance of a recurring meeting has the **duration** field (3) set to the duration of an instance in the series. This field (3) is NULL for all other instances in the series. For standalone meetings tied to workspaces, this field (3) is NULL.

**ntext1:** For the default meeting instance, this is the XML fragment describing the recurrence rule for the recurring meeting series. For individual meeting instances, this field (3) MUST be NULL.

**ntext2:** Any recurrence rule specified for the recurring meeting. When specified, this field (3) is set for the default meeting instance and for other instances, this is NULL.

**ntext3:** This property returns a rule or repeating pattern defined for an exception to the specified recurring meeting.

**datetime5:** If the series has been suppressed, this is the date in UTC it has been suppressed until. When this field (3) is NULL, it indicates that the series has not been suppressed.

**int3:** The **integer** representing the time zone used when creating this meeting.

### 3.1.5.16 proc\_GetDocComments

The **proc\_GetDocComments** stored procedure is called to return the set of Web discussion comments associated with a document. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetDocComments (
    @DocSiteId          uniqueidentifier,
    @DocFullUrl         nvarchar(260),
    @RequestGuid        uniqueidentifier = NULL OUTPUT
);
```

**@DocSiteId:** The site collection identifier of the site collection containing the specified document.

**@DocFullUrl:** The store-relative URL to the document.

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

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

**Result Sets:** MUST return one result set.

#### 3.1.5.16.1 GetDocComments Result Set

Returns the set of Web discussion comments associated with this document. This result set contains one row (3) for each associated Web discussion comment. The T-SQL syntax for the result set is as follows:

Id	int,
Parent	int,
CommentId	nvarchar(255),
Bookmark	nvarchar(127),
Author	nvarchar(255),
tp_Login	nvarchar(255),
Subject	nvarchar(255),
Created	datetime,
Status	smallint,

Comment                          ntext;

**Id:** The identifier of the Web discussion comment.

**Parent:** The identifier of the parent Web discussion comment. If the parent Web discussion comment does not exist, this MUST be zero ("0").

**CommentId:** The identifier, specified by the protocol client, of the Web discussion comment.

**Bookmark:** A protocol client specified reference to the place in the document to which the Web discussion comment refers.

**Author:** A protocol client specified name for the user that added the Web discussion comment.

**tp\_Login:** The **login name** of the user that added the Web discussion comment.

**Subject:** The subject of the Web discussion comment.

**Created:** The time in UTC when this Web discussion comment was created.

**Status:** A Web discussion comment status flag (section [2.2.2.3](#)) that indicates the status of a Web discussion comment.

**Comment:** The body text of the Web discussion comment.

### 3.1.5.17 **proc\_GetEventDataAndSubscriptionFilters**

The **proc\_GetEventDataAndSubscriptionFilters** stored procedure is called to return events (2) and alert subscriptions and mark the returned events (2) and alert subscriptions as having been processed. The processed alert subscriptions and events (2) are not returned in subsequent calls to this stored procedure. The T-SQL syntax, as follows:

```
PROCEDURE proc_GetEventDataAndSubscriptionFilters (  
    @MaxBatchSize        bigint,  
    @DtLast             datetime,  
    @RequestGuid        uniqueidentifier = NULL OUTPUT  
);
```

**@MaxBatchSize:** The maximum number of events (2) to process in this batch.

**@DtLast:** The end datetime in UTC of the event (2) batch.

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

**Return Values:** MUST return an **integer** value that MUST be zero.

**Result Sets:** MUST return three result sets as defined in the following subsections.

#### 3.1.5.17.1 **Events Result Set**

The **Events** result set MUST be returned. The T-SQL syntax for the result set is as follows:

```
ListId                      uniqueidentifier  
Id                          bigint,  
EventData                  image,
```

EventType	int,
ItemId	int,
ItemName	nvarchar(255),
ItemFullUrl	nvarchar(260),
ModifiedBy	nvarchar(255),
TimeLastModified	datetime,
ACL	image,
SiteId	uniqueidentifier;

**ListId:** The list identifier of the list (1) that the event (2) is related to. This value MUST NOT be NULL.

**Id:** The identifier of the event (2). This value MUST NOT be NULL.

**EventData:** Contains implementation-specific event (2) data significant to the front-end Web server but otherwise opaque to the back-end database server.

**EventType:** An alert (1) event (2) type (section [2.2.2.1](#)) that specifies an **integer** mask for the types of events (2) on which to fire the notification. This value MUST NOT be NULL.

**ItemId:** The identifier of the list item from the list (1) specified by **ListId** that is associated with the event (2). This MUST NOT be NULL.

**ItemName:** A **string** that represents the name of the list item chosen by the application. This parameter MUST be set to NULL if the application does not use this information.

**ItemFullUrl:** The URL to the list item that is associated with the event (2), or the URL to the site (2), or NULL if the application already specifies the list item using *@ItemId*.

**ModifiedBy:** A string that specifies the login name of a **security principal(2)** who added this event (2).

**TimeLastModified:** A time stamp in UTC that specifies the time when this event (2) happened. This value MUST NOT be NULL.

**ACL:** A byte array in the ACL format. If this parameter is NULL, the ACL will be inferred from the *@ItemFullUrl*.

**SiteId:** The site collection identifier for the site collection under which the event (2) has occurred. This value MUST NOT be NULL.

### 3.1.5.17.2 Subscriptions Result Set

The **Subscriptions** result set returns the alert subscriptions for the site (2) and list (1) of the events (2) returned by the **Events** result set. The **Subscriptions** result set MUST be returned. The T-SQL syntax is as follows:

Id	uniqueidentifier,
BinaryFilter	varbinary(1024),
ListId	uniqueidentifier,
ItemId	int,
AlertType	int,
EventType	int,
{NotifyFreq}	int,
WebTimeZone	smallint,
WebId	uniqueidentifier,

UserId	int,
SiteId	uniqueidentifier,
SiteUrl	nvarchar(136),
WebUrl	nvarchar(256),
Properties	ntext,
DeliveryChannel	int;

**Id:** The alert subscription identifier of this alert subscription. The value MUST NOT be NULL.

**BinaryFilter:** The compiled binary syntax query in **Collaborative Application Markup Language (CAML)** for a filter to apply to the event (2) data of the event (2) associated with this alert subscription.

**ListId:** The list identifier for the list (1) with which this alert subscription is associated. The value MUST NOT be NULL.

**ItemId:** The **item identifier** of the item with which this alert subscription is associated. This MUST NOT be NULL if the alert (1) type (section [2.2.1.3](#)) of the alert subscription is item and MUST be NULL if the alert (1) type of the alert subscription is list (1).

**AlertType:** An **integer** whose lower eight bits specifies the type of the alert (1) as specified in section [2.2.1.3](#). This MUST also contain the special alert (1) flags (section [2.2.2.2](#)) for an always notify alert or a system alert.

**EventType:** An alert (1) event (2) type (section [2.2.2.1](#)) that specifies an **integer** mask for the types of events (2) on which to fire the notification. This value MUST NOT be NULL.

**{NotifyFreq}:** A notification frequency type (section [2.2.1.6](#)) that specifies an **integer** indicating the frequency of the alert (1) notification. This value MUST NOT be NULL.

**WebTimeZone:** The time zone of the site (2) with which this alert subscription is associated. The value MUST NOT be NULL.

**WebId:** The site identifier for the site (2) with which this alert subscription is associated. The value MUST NOT be NULL.

**UserId:** The user identifier for the user with which this alert subscription is associated. The value MUST NOT be NULL.

**SiteId:** The site collection identifier for the site collection with which this alert subscription is associated. The value MUST NOT be NULL.

**SiteUrl:** The URL of the site collection with which this alert subscription is associated. The value MUST NOT be NULL.

**WebUrl:** The URL of the site (2) with which this alert subscription is associated. The value MUST NOT be NULL.

**Properties:** An XML blob representing the properties of the alert (1).

**DeliveryChannel:** Specifies the delivery channel for the alert subscription as defined by the alert (1) delivery channel type (section [2.2.1.1](#)).



### 3.1.5.17.3 UTCTime Result Set

The **UTCTime** result set returns the start and end identifiers of the events (2) marked for processing in this call to the stored procedure. The **UTCTime** result set MUST be returned and MUST have one row (3). The T-SQL syntax is as follows:

```
{IdStart}          bigint,  
{IdEnd}           bigint;
```

**{IdStart}**: The identifier of the first event (2) in this batch.

**{IdEnd}**: The identifier of the last event (2) in this batch.

### 3.1.5.18 proc\_GetExceptionIDs

The **proc\_GetExceptionIDs** stored procedure retrieves the instances of a recurring meeting series that are exceptions. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetExceptionIDs(  
    @ListID          uniqueidentifier,  
    @ViewStart       datetime,  
    @ViewEnd         datetime,  
    @EventTypeColName nvarchar(64),  
    @EventTypeRowOrdinal int = 0,  
    @UIDColName      nvarchar(64),  
    @UIDRowOrdinal  tinyint = 0,  
    @RecurrenceIDColName nvarchar(64),  
    @RecurrenceIDRowOrdinal tinyint = 0,  
    @StartDateColName nvarchar(64),  
    @StartDateRowOrdinal tinyint = 0,  
    @EndDateColName  nvarchar(64),  
    @EndDateRowOrdinal tinyint = 0,  
    @RequestGuid     uniqueidentifier = NULL OUTPUT  
);
```

**@ListId**: A GUID of a **recurrence list**.

**@ViewStart**: A datetime field (3) corresponding to the start date of the recurring meeting.

**@ViewEnd**: A datetime field (3) corresponding to the End Date of the recurring meeting.

**@EventTypeColName**: The name of the column (1) in the **UserData** view that corresponds to the event (2) type. For more details about **UserData** view, see [\[MS-WSSFO2\]](#) section 2.2.7.8.

**@EventTypeRowOrdinal**: The ordinal of a list item that MUST contain the event (2) type information specified inside the column (1) identified by the parameter *@EventTypeColName*.

**@UIDColName**: The name of the column (1) in the **UserData** view that corresponds to the unique identifier of the recurring meeting series.

**@UIDRowOrdinal**: The name of the row (3) ordinal in the **UserData** view that MUST contain the unique identifier of the recurring meeting series in the column (1) specified by the *@UIDColName* parameter.

**@RecurrenceIDColName**: The name of the column (1) in the **UserData** view that corresponds to the recurrence identifier of the recurring meeting series.

**@RecurrenceIDRowOrdinal:** The ordinal of list item that MUST contain the recurrence identifier specified inside the column (1) identified by the parameter *@RecurrenceIDColName*.

**@StartDateColName:** The name of the column (1) in the **UserData** view that corresponds to the start date of the recurring meeting series. For more details about **UserData** view, see [\[MS-WSSFO2\]](#) section 2.2.7.8.

**@StartDateRowOrdinal:** The ordinal of list item that MUST contain the start date information specified inside the column (1) identified by the parameter *@StartDateColName*.

**@EndDateColName:** The name of the column (1) in the **UserData** view that corresponds to the End Date of the recurring meeting series.

**@EndDateRowOrdinal:** The ordinal of list item that MUST contain the End Date information specified inside the column (1) identified by the parameter *@EndDateColName*.

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

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

**Result Sets:** MUST return one result set when the column (1) names and row (3) ordinals specified for the input parameters exist in the **UserData** view and map to a valid entity in the **UserData** view.

### 3.1.5.18.1 Exceptions Result Set

The **Exceptions** result set returns the identifier information for a recurring meeting and a recurrence identifier to specifically identify a meeting instance. The **Exceptions** result set contains one or more rows (3) if the specified input parameters map to valid meeting series exceptions in the **UserData** view. For more details about **UserData** view, see [\[MS-WSSFO2\]](#) section 2.2.7.8.

The **Exceptions** result set is empty if the column (1) names and ordinals specified as input parameters exist in the **UserData** view but do not map to any valid meeting series exceptions.

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

```
{UID}                uniqueidentifier,  
{RecurrenceID}      datetime;
```

**{UID}:** The name of the column (1) is specified by the *@UIDColName* parameter. The identifier of the recurring meeting.

**{RecurrenceID}:** The name of the column (1) is specified by the *@RecurrenceIDColName* parameter. The datetime information that makes up the recurrence identifier.

### 3.1.5.19 proc\_GetFutureExceptionIDsForUID

The **proc\_GetFutureExceptionIDsForUID** stored procedure returns all the future instances in a recurring meeting series that are exceptions. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetFutureExceptionIDsForUID(  
    @ListID                uniqueidentifier,  
    @DTStamp               datetime,  
    @UID                   uniqueidentifier,  
    @EventTypeColName      nvarchar(64),  
    -
```

```

@EventTypeRowOrdinal    tinyint = 0,
@UIDColName             nvarchar(64),
@UIDRowOrdinal          tinyint = 0,
@RecurrenceIDColName   nvarchar(64),
@RecurrenceIDRowOrdinal tinyint = 0,
@StartDateColName      nvarchar(64),
@StartDateRowOrdinal   tinyint = 0,
@RequestGuid            uniqueidentifier = NULL OUTPUT
);

```

**@ListId:** A UID of a recurrence list.

**@DTStamp:** The procedure will return all the instance exceptions occurring after the datetime specified in this parameter.

**@UID:** This is the unique identifier of the recurring meeting series.

**@EventTypeColName:** The name of the column (1) in the **UserData** view that corresponds to the event (2) type. For more details about **UserData** view, see [\[MS-WSSFO2\]](#) section 2.2.7.8.

**@EventTypeRowOrdinal:** The ordinal of list item that MUST contain the event (2) type information specified inside the column (1) identified by the parameter *@EventTypeColName*.

**@UIDColName:** The name of the column (1) in the **UserData** view that corresponds to the unique identifier of the recurring meeting series.

**@UIDRowOrdinal:** The name of the row (3) ordinal in the **UserData** view that MUST contain the unique identifier of the instance of the recurring meeting in the column (1) specified by the *@UIDColName* parameter.

**@RecurrenceIDColName:** The name of the column (1) in the **UserData** view that corresponds to the recurrence identifier of the recurring meeting series.

**@RecurrenceIDRowOrdinal:** The ordinal of list item that MUST contain the recurrence identifier specified inside the column (1) identified by the parameter *@RecurrenceIDColName*.

**@StartDateColName:** The name of the column (1) in the **UserData** view that corresponds to the start date of the recurring meeting series.

**@StartDateRowOrdinal:** The ordinal of list item that MUST contain the start date information specified inside the column (1) identified by the parameter *@StartDateColName*.

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

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

**Result Sets:** MUST return one result set when the column (1) names and row (3) ordinals specified for the input parameters exist in the **UserData** view and map to a valid entity in the **UserData** view.

### 3.1.5.19.1 Exceptions Result Set

The **Exceptions** result set returns the **list item identifier** for a meeting instance and the **RecurrenceID** that identifies the specific meeting instance. The **Exceptions** result set contains one or more rows (3) if the specified input parameters map to valid meeting series exceptions in the **UserData** view. The Exceptions result set is empty if the column (1) names and ordinals specified

as input parameters, exist in the **UserData** view but do not map to any valid meeting series exceptions. (For more details about **UserData** view, see [\[MS-WSSFO2\]](#) section 2.2.7.8.) The T-SQL syntax for the result set is as follows:

```
tp_ID          int,  
{RecurrenceID}  datetime;
```

**tp\_ID:** The list item identifier of the meeting instance.

**{RecurrenceID}:** The datetime information that makes up the recurrence identifier. The name of the column (1) is specified the **@RecurrenceIDColName** parameter.

### 3.1.5.20 **proc\_GetItemCountPerInstance**

The **proc\_GetItemCountPerInstance** stored procedure is called to retrieve the number of items within a given list (1) associated with the given **InstanceID**. Items in the given list (1) with a NULL **InstanceID** are also included in the count. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetItemCountPerInstance(  
    @ListID          uniqueidentifier,  
    @InstanceID      int,  
    @ItemCount       int OUTPUT,  
    @RequestGuid     uniqueidentifier = NULL OUTPUT  
);
```

**@ListId:** Specifies the identifier of the list (1) in which the items exist.

**@InstanceID:** Specifies the identifier of the instance for which the caller is requesting the item count.

**@ItemCount:** The number of items in the specified list (1) with the specified **InstanceID** or NULL. This value is filled and returned to the caller.

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

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

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

### 3.1.5.21 **proc\_GetListAndChildrenNSInfo**

The **proc\_GetListAndChildrenNSInfo** stored procedure is called to retrieve the subfolders of a root folder for a list (1), or the subfolders of a folder, the sort behavior of these subfolders MUST NOT be like the sort behavior of files. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetListAndChildrenNSInfo(  
    @SiteId          uniqueidentifier,  
    @WebId           uniqueidentifier,  
    @ListId          uniqueidentifier,  
    @FolderId        uniqueidentifier,  
    @GetList         bit = 0,  
    @GetSubFolders   bit = 0,  
    @Lcid            bit = 0,  
    @RequestGuid     uniqueidentifier = NULL OUTPUT
```

);

**@SiteId:** Specifies the identifier of the site collection in which the list (1) exists.

**@WebId:** Specifies the identifier of the site (2) in which the list (1) exists.

**@ListId:** Specifies the identifier of the list (1) in which the list (1) containing the root folder and subfolders exist.

**@FolderId:** Specifies the identifier of the folder in which the subfolders exist. If *@GetList* is "1", this input parameter MUST be ignored.

**@GetList:** If this parameter is "1", a **ListRootFolderNSInfo** result set MUST be returned. The default value is zero ("0").

**@GetSubFolders:** If this parameter is "1", a **ChildFoldersNSInfo** result set MUST be returned. The default value is zero ("0").

**@Lcid:** The LCID for the list (1) specified by the *@WebId* and *@ListId* parameters.

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

**Return Values:** Returns an **integer** return code that MUST be zero.

**Result Sets:** MUST return zero, one, or two result sets based on the values of the input parameters *@GetList* and *@GetSubFolders*.

Input Parameter	Returned Result Set
<i>@GetList</i> = "1"	One <b>ListRootFolderNSInfo</b> result set
<i>@GetSubFolders</i> = "1"	One <b>ChildFoldersNSInfo</b> result set

### 3.1.5.21.1 ListRootFolderNSInfo Result Set

The **ListRootFolderNSInfo** result set contains a row (3) set containing the information about the list (1), its root folder and the permissions on its root folder. The **ListRootFolderNSInfo** result set MUST be returned when *@GetList* is set to "1" and it MUST contain one row (3) if the list (1) can be found. It MUST contain no rows (3) if the specified list (1) doesn't exist. The T-SQL syntax for the result set is as follows:

```
tp_Title          nvarchar(255),
tp_Id             uniqueidentifier,
FullUrl           nvarchar(260),
tp_RootFolder     uniqueidentifier,
tp_ImageUrl       nvarchar(255),
tp_BaseType       int,
tp_ServerTemplate int,
FolderChildCount int,
UserResource      nvarchar(256) or nvarchar(max),
Acl               image,
AnonymousPermMask bigint;
```

**tp\_Title:** The title of the list (1).

**tp\_Id:** The list identifier of the list (1).

**FullUrl:** The complete store-relative URL for the root folder for the list (1).

**tp\_RootFolder:** The identifier of the root folder for the list (1).

**tp\_ImageUrl:** The URL of the image used to represent the list (1).

**tp\_BaseType:** The base type of the list (1), as defined in [\[MS-WSSFO2\]](#) section 2.2.3.11.

**tp\_ServerTemplate:** The **List Server Template** enumeration value of the list template that defines the base structure of the list (1). See [\[MS-WSSFO2\]](#) section 2.2.3.12, for more detail.

**FolderChildCount:** The number of child objects (1) of the root folder.

**UserResource:** The localized list (1) title specified by *@WebId*, *@ListId* and *@Lcid*.

**Acl:** The binary serialization of the ACL for the root folder. The format is specified in [\[MS-WSSFO2\]](#) section 2.2.4.6.

**AnonymousPermMask:** The permissions mask that applies to an anonymous user of the root folder. The format is specified in [\[MS-WSSFO2\]](#) section 2.2.2.14.

### 3.1.5.21.2 ChildFoldersNSInfo Result Set

The **ChildFoldersNSInfo** result set returns a rowset containing the subfolders of the root folder for the specified list (1) if *@GetSubFolders* is "1". The **ChildFoldersNSInfo** result set MUST be returned when the input parameter *@GetSubFolders* is "1". The **ChildFoldersNSInfo** result set MUST contain zero or more rows (3). The T-SQL syntax for the result set is as follows:

```
LeafName          nvarchar(128),
Id                uniqueidentifier,
{FullUrl}         nvarchar(260),
FolderChildCount int,
ListId           uniqueidentifier,
Acl              image,
AnonymousPermMask bigint;
```

**LeafName:** The store-relative form leaf name of the subfolder.

**Id:** The identifier of the subfolder.

**{FullUrl}:** The complete store-relative URL for the root folder for the list (1).

**FolderChildCount:** The number of the subfolder.

**ListId:** The identifier of the list (1) containing the folder.

**Acl:** The binary serialization of the ACL for the folder. The format is specified in [\[MS-WSSFO2\]](#) section 2.2.4.6.

**AnonymousPermMask:** The permissions mask that applies to an anonymous user of the subfolder. The format is specified in [\[MS-WSSFO2\]](#) section 2.2.2.14.

### 3.1.5.22 proc\_GetMatchingSubscriptionsData

The **proc\_GetMatchingSubscriptionsData** stored procedure is called to return alert **subscriptions**.

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

```
PROCEDURE proc_GetMatchingSubscriptionsData (  
    @f001                uniqueidentifier = NULL,  
    @f002                uniqueidentifier = NULL,  
    @f003                uniqueidentifier = NULL,  
    @f004                uniqueidentifier = NULL,  
    @f005                uniqueidentifier = NULL,  
    @f006                uniqueidentifier = NULL,  
    @f007                uniqueidentifier = NULL,  
    @f008                uniqueidentifier = NULL,  
    @f009                uniqueidentifier = NULL,  
    @f010                uniqueidentifier = NULL,  
    @f011                uniqueidentifier = NULL,  
    @f012                uniqueidentifier = NULL,  
    @f013                uniqueidentifier = NULL,  
    @f014                uniqueidentifier = NULL,  
    @f015                uniqueidentifier = NULL,  
    @f016                uniqueidentifier = NULL,  
    @f017                uniqueidentifier = NULL,  
    @f018                uniqueidentifier = NULL,  
    @f019                uniqueidentifier = NULL,  
    @f020                uniqueidentifier = NULL,  
    @f021                uniqueidentifier = NULL,  
    @f022                uniqueidentifier = NULL,  
    @f023                uniqueidentifier = NULL,  
    @f024                uniqueidentifier = NULL,  
    @f025                uniqueidentifier = NULL,  
    @f026                uniqueidentifier = NULL,  
    @f027                uniqueidentifier = NULL,  
    @f028                uniqueidentifier = NULL,  
    @f029                uniqueidentifier = NULL,  
    @f030                uniqueidentifier = NULL,  
    @f031                uniqueidentifier = NULL,  
    @f032                uniqueidentifier = NULL,  
    @f033                uniqueidentifier = NULL,  
    @f034                uniqueidentifier = NULL,  
    @f035                uniqueidentifier = NULL,  
    @f036                uniqueidentifier = NULL,  
    @f037                uniqueidentifier = NULL,  
    @f038                uniqueidentifier = NULL,  
    @f039                uniqueidentifier = NULL,  
    @f040                uniqueidentifier = NULL,  
    @f041                uniqueidentifier = NULL,  
    @f042                uniqueidentifier = NULL,  
    @f043                uniqueidentifier = NULL,  
    @f044                uniqueidentifier = NULL,  
    @f045                uniqueidentifier = NULL,  
    @f046                uniqueidentifier = NULL,  
    @f047                uniqueidentifier = NULL,  
    @f048                uniqueidentifier = NULL,  
    @f049                uniqueidentifier = NULL,  
    @f050                uniqueidentifier = NULL,  
    @f051                uniqueidentifier = NULL,
```

@f052	uniqueidentifier = NULL,
@f053	uniqueidentifier = NULL,
@f054	uniqueidentifier = NULL,
@f055	uniqueidentifier = NULL,
@f056	uniqueidentifier = NULL,
@f057	uniqueidentifier = NULL,
@f058	uniqueidentifier = NULL,
@f059	uniqueidentifier = NULL,
@f060	uniqueidentifier = NULL,
@f061	uniqueidentifier = NULL,
@f062	uniqueidentifier = NULL,
@f063	uniqueidentifier = NULL,
@f064	uniqueidentifier = NULL,
@f065	uniqueidentifier = NULL,
@f066	uniqueidentifier = NULL,
@f067	uniqueidentifier = NULL,
@f068	uniqueidentifier = NULL,
@f069	uniqueidentifier = NULL,
@f070	uniqueidentifier = NULL,
@f071	uniqueidentifier = NULL,
@f072	uniqueidentifier = NULL,
@f073	uniqueidentifier = NULL,
@f074	uniqueidentifier = NULL,
@f075	uniqueidentifier = NULL,
@f076	uniqueidentifier = NULL,
@f077	uniqueidentifier = NULL,
@f078	uniqueidentifier = NULL,
@f079	uniqueidentifier = NULL,
@f080	uniqueidentifier = NULL,
@f081	uniqueidentifier = NULL,
@f082	uniqueidentifier = NULL,
@f083	uniqueidentifier = NULL,
@f084	uniqueidentifier = NULL,
@f085	uniqueidentifier = NULL,
@f086	uniqueidentifier = NULL,
@f087	uniqueidentifier = NULL,
@f088	uniqueidentifier = NULL,
@f089	uniqueidentifier = NULL,
@f090	uniqueidentifier = NULL,
@f091	uniqueidentifier = NULL,
@f092	uniqueidentifier = NULL,
@f093	uniqueidentifier = NULL,
@f094	uniqueidentifier = NULL,
@f095	uniqueidentifier = NULL,
@f096	uniqueidentifier = NULL,
@f097	uniqueidentifier = NULL,
@f098	uniqueidentifier = NULL,
@f099	uniqueidentifier = NULL,
@f100	uniqueidentifier = NULL,
@f101	uniqueidentifier = NULL,
@f102	uniqueidentifier = NULL,
@f103	uniqueidentifier = NULL,
@f104	uniqueidentifier = NULL,
@f105	uniqueidentifier = NULL,
@f106	uniqueidentifier = NULL,
@f107	uniqueidentifier = NULL,
@f108	uniqueidentifier = NULL,
@f109	uniqueidentifier = NULL,
@f110	uniqueidentifier = NULL,



@f111	uniqueidentifier = NULL,
@f112	uniqueidentifier = NULL,
@f113	uniqueidentifier = NULL,
@f114	uniqueidentifier = NULL,
@f115	uniqueidentifier = NULL,
@f116	uniqueidentifier = NULL,
@f117	uniqueidentifier = NULL,
@f118	uniqueidentifier = NULL,
@f119	uniqueidentifier = NULL,
@f120	uniqueidentifier = NULL,
@f121	uniqueidentifier = NULL,
@f122	uniqueidentifier = NULL,
@f123	uniqueidentifier = NULL,
@f124	uniqueidentifier = NULL,
@f125	uniqueidentifier = NULL,
@f126	uniqueidentifier = NULL,
@f127	uniqueidentifier = NULL,
@f128	uniqueidentifier = NULL,
@f129	uniqueidentifier = NULL,
@f130	uniqueidentifier = NULL,
@f131	uniqueidentifier = NULL,
@f132	uniqueidentifier = NULL,
@f133	uniqueidentifier = NULL,
@f134	uniqueidentifier = NULL,
@f135	uniqueidentifier = NULL,
@f136	uniqueidentifier = NULL,
@f137	uniqueidentifier = NULL,
@f138	uniqueidentifier = NULL,
@f139	uniqueidentifier = NULL,
@f140	uniqueidentifier = NULL,
@f141	uniqueidentifier = NULL,
@f142	uniqueidentifier = NULL,
@f143	uniqueidentifier = NULL,
@f144	uniqueidentifier = NULL,
@f145	uniqueidentifier = NULL,
@f146	uniqueidentifier = NULL,
@f147	uniqueidentifier = NULL,
@f148	uniqueidentifier = NULL,
@f149	uniqueidentifier = NULL,
@f150	uniqueidentifier = NULL,
@f151	uniqueidentifier = NULL,
@f152	uniqueidentifier = NULL,
@f153	uniqueidentifier = NULL,
@f154	uniqueidentifier = NULL,
@f155	uniqueidentifier = NULL,
@f156	uniqueidentifier = NULL,
@f157	uniqueidentifier = NULL,
@f158	uniqueidentifier = NULL,
@f159	uniqueidentifier = NULL,
@f160	uniqueidentifier = NULL,
@f161	uniqueidentifier = NULL,
@f162	uniqueidentifier = NULL,
@f163	uniqueidentifier = NULL,
@f164	uniqueidentifier = NULL,
@f165	uniqueidentifier = NULL,
@f166	uniqueidentifier = NULL,
@f167	uniqueidentifier = NULL,
@f168	uniqueidentifier = NULL,
@f169	uniqueidentifier = NULL,

@f170	uniqueidentifier = NULL,
@f171	uniqueidentifier = NULL,
@f172	uniqueidentifier = NULL,
@f173	uniqueidentifier = NULL,
@f174	uniqueidentifier = NULL,
@f175	uniqueidentifier = NULL,
@f176	uniqueidentifier = NULL,
@f177	uniqueidentifier = NULL,
@f178	uniqueidentifier = NULL,
@f179	uniqueidentifier = NULL,
@f180	uniqueidentifier = NULL,
@f181	uniqueidentifier = NULL,
@f182	uniqueidentifier = NULL,
@f183	uniqueidentifier = NULL,
@f184	uniqueidentifier = NULL,
@f185	uniqueidentifier = NULL,
@f186	uniqueidentifier = NULL,
@f187	uniqueidentifier = NULL,
@f188	uniqueidentifier = NULL,
@f189	uniqueidentifier = NULL,
@f190	uniqueidentifier = NULL,
@f191	uniqueidentifier = NULL,
@f192	uniqueidentifier = NULL,
@f193	uniqueidentifier = NULL,
@f194	uniqueidentifier = NULL,
@f195	uniqueidentifier = NULL,
@f196	uniqueidentifier = NULL,
@f197	uniqueidentifier = NULL,
@f198	uniqueidentifier = NULL,
@f199	uniqueidentifier = NULL,
@f200	uniqueidentifier = NULL,
@f201	uniqueidentifier = NULL,
@f202	uniqueidentifier = NULL,
@f203	uniqueidentifier = NULL,
@f204	uniqueidentifier = NULL,
@f205	uniqueidentifier = NULL,
@f206	uniqueidentifier = NULL,
@f207	uniqueidentifier = NULL,
@f208	uniqueidentifier = NULL,
@f209	uniqueidentifier = NULL,
@f210	uniqueidentifier = NULL,
@f211	uniqueidentifier = NULL,
@f212	uniqueidentifier = NULL,
@f213	uniqueidentifier = NULL,
@f214	uniqueidentifier = NULL,
@f215	uniqueidentifier = NULL,
@f216	uniqueidentifier = NULL,
@f217	uniqueidentifier = NULL,
@f218	uniqueidentifier = NULL,
@f219	uniqueidentifier = NULL,
@f220	uniqueidentifier = NULL,
@f221	uniqueidentifier = NULL,
@f222	uniqueidentifier = NULL,
@f223	uniqueidentifier = NULL,
@f224	uniqueidentifier = NULL,
@f225	uniqueidentifier = NULL,
@f226	uniqueidentifier = NULL,
@f227	uniqueidentifier = NULL,
@f228	uniqueidentifier = NULL,

```

@f229                uniqueidentifier = NULL,
@f230                uniqueidentifier = NULL,
@f231                uniqueidentifier = NULL,
@f232                uniqueidentifier = NULL,
@f233                uniqueidentifier = NULL,
@f234                uniqueidentifier = NULL,
@f235                uniqueidentifier = NULL,
@f236                uniqueidentifier = NULL,
@f237                uniqueidentifier = NULL,
@f238                uniqueidentifier = NULL,
@f239                uniqueidentifier = NULL,
@f240                uniqueidentifier = NULL,
@f241                uniqueidentifier = NULL,
@f242                uniqueidentifier = NULL,
@f243                uniqueidentifier = NULL,
@f244                uniqueidentifier = NULL,
@f245                uniqueidentifier = NULL,
@f246                uniqueidentifier = NULL,
@f247                uniqueidentifier = NULL,
@f248                uniqueidentifier = NULL,
@f249                uniqueidentifier = NULL,
@f250                uniqueidentifier = NULL,
@f251                uniqueidentifier = NULL,
@f252                uniqueidentifier = NULL,
@f253                uniqueidentifier = NULL,
@f254                uniqueidentifier = NULL,
@f255                uniqueidentifier = NULL,
@f256                uniqueidentifier = NULL,
@RequestGuid        uniqueidentifier = NULL OUTPUT
);

```

**@fnnn:** The alert subscription identifier for which alert subscription data is to be returned.

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

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

**Result Sets:** **MUST** return one result set, as defined in the following subsection.

### 3.1.5.22.1 MatchingSubscriptionsData Result Set

The **MatchingSubscriptionsData** result set contains the alert subscriptions for any of the identifiers defined by the **@fnnn** arguments. The **MatchingSubscriptionsData** result set **MUST** be returned. The T-SQL syntax for the result set is as follows:

```

WebId                uniqueidentifier,
UserEmail            nvarchar(255),
UserId              int,
SiteUrl             nvarchar(136),
WebUrl             nvarchar(256),
ListUrl            nvarchar(256),
WebTitle           nvarchar(255),
WebLanguage         int,
WebLocale          int,
WebTimeZone        smallint,
WebTime24          bit,
WebCalendarType    smallint,

```

WebAdjustHijriDays	smallint,
ListId	uniqueidentifier,
ListTitle	nvarchar(255),
ListBaseType	int,
ListServerTemplate	int,
Id	uniqueidentifier,
ItemId	int,
EventType	int,
Binaryfilter	varbinary(1024),
Properties	ntext,
AlertTitle	nvarchar(1000),
AlertType	int,
AlertTemplateName	nvarchar(255),
DeliveryChannel	int;

**WebId:** The site identifier with which this alert subscription is associated. The value MUST NOT be NULL.

**UserEmail:** The e-mail address of the user with which this alert subscription is associated. The value MUST NOT be NULL.

**UserId:** The user identifier with which this alert subscription is associated. The value MUST NOT be NULL.

**SiteUrl:** The URL of the site collection with which this alert subscription is associated. The value MUST NOT be NULL.

**WebUrl:** The URL of the site (2) with which this alert subscription is associated. The value MUST NOT be NULL.

**ListUrl:** The URL of the list (1) with which this alert subscription is associated. The value MUST NOT be NULL.

**WebTitle:** The title of the site (2) with which this alert subscription is associated. The value MUST NOT be NULL.

**WebLanguage:** The language of the site (2) with which this alert subscription is associated. It is an **integer**. The value MUST NOT be NULL.

**WebLocale:** The locale of the site (2) with which this alert subscription is associated. The value MUST NOT be NULL.

**WebTimeZone:** The time zone of the site (2) with which this alert subscription is associated. The value MUST NOT be NULL.

**WebTime24:** The time format that specifies whether the time displayed for the site (2) with which this alert subscription is associated is in 12-hour or 24-hour format. The value MUST NOT be NULL.

**WebCalendarType:** Contains calendar type (for non-Gregorian calendars) of the site (2) with which this alert subscription is associated. The value MUST NOT be NULL.

**WebAdjustHijriDays:** The number of days to extend or reduce the current month in Hijri calendars on the site (2) with which this alert subscription is associated. The value MUST NOT be NULL.

**ListId:** The list identifier for the list (1) with which this alert subscription is associated. The value MUST NOT be NULL.

**ListTitle:** The title of the list (1) with which this alert subscription is associated. The value MUST NOT be NULL.

**ListBaseType:** The base type of the list (1) with which this alert subscription is associated. The value MUST NOT be NULL.

**ListServerTemplate:** The list template for the list (1) with which this alert subscription is associated. The value MUST NOT be NULL.

**Id:** The identifier of this alert subscription. The value MUST NOT be NULL.

**ItemId:** The item identifier with which this alert subscription is associated. This MUST NOT be NULL if the alert (1) type (section [2.2.1.3](#)) of the alert subscription is Item alert and MUST be NULL if the alert (1) type of the alert subscription is List alert.

**EventType:** An alert (1) event (2) type (section [2.2.2.1](#)) that specifies an **integer** mask for the types of events (2) on which to send the notification. This value MUST NOT be NULL.

**Binaryfilter:** Contains compiled binary syntax query in **CAML** for a filter to apply to the alert (1) associated with this alert subscription.

**Properties:** An XML blob representing the properties of the alert (1).

**AlertTitle:** The title of the alert (1) of the alert subscription.

**AlertType:** An **integer** whose lower eight bits specifies the type of the alert (1) as specified in section [2.2.1.3](#). This MUST also contain the special alert (1) flags (section [2.2.2.2](#)) for an always notify alert or a system alert.

**AlertTemplateName:** The name of the alert template of the alert subscription.

**DeliveryChannel:** Specifies the delivery channel for the alert subscription as defined by the alert (1) delivery channel type (section [2.2.1.1](#)).

### 3.1.5.23 **proc\_GetMeetingInstanceDataForICal**

The **proc\_GetMeetingInstanceDataForICal** stored procedure is called to retrieve the data needed to construct **iCalendar** information from a specified meeting instance in a specified Meeting Workspace site.

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

```
PROCEDURE proc_GetMeetingInstanceDataForICal (
    @WebId                uniqueidentifier,
    @InstanceID           int,
    @@OrganizerID         int OUTPUT,
    @@UID                  nvarchar(255) OUTPUT,
    @@DTStartUTC           datetime OUTPUT,
    @@DTEndUTC             datetime OUTPUT,
    @@DTStampUTC           datetime OUTPUT,
    @@Sequence             int OUTPUT,
    @RequestGuid           uniqueidentifier = NULL OUTPUT
);
```

**@WebId:** Specifies the identifier of the site (2) in which the meeting series exists. *@WebId* MUST NOT be NULL.

**@InstanceID:** Specifies the identifier of the meeting instance for which data is being requested. This MUST NOT be NULL.

**@@OrganizerID:** Specifies the identifier of the meeting organizer of this meeting instance. This parameter is returned to the caller.

**@@UID:** The unique identifier for this meeting instance. This parameter is returned to the caller.

**@@DTStartUTC:** The starting datetime for the meeting instance, in UTC. This parameter is returned to the caller.

**@@DTEndUTC:** The ending datetime for the meeting instance, in UTC. This parameter is returned to the caller.

**@@DTStampUTC:** The timestamp datetime for the creation of the meeting instance, in UTC. This parameter is returned to the caller.

**@@Sequence:** The revision **sequence number (1)** of the meeting instance. This property defines the revision sequence number (1) of the meeting instance within a sequence of revisions. This parameter is returned to the caller.

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

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

Value	Description
"0"	Successful completion.
"2"	No meeting instances found for the given <i>@WebId</i> and <i>@InstanceID</i> .
"13"	Two possible failure conditions: <ul style="list-style-type: none"><li>▪ No meetings list (1) found on the Meeting Workspace site specified by <i>@WebId</i>.</li><li>▪ <i>@@OrganizerID</i> or <i>@@UID</i> for the meeting instance is NULL.</li></ul>

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

### 3.1.5.24 **proc\_GetRecurrenceSeriesData**

The **proc\_GetRecurrenceSeriesData** stored procedure is called to get Series UID, start date, and **MasterSeriesID** for the given meeting series list (1), meeting series item, and meeting calendar event type.

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

```
PROCEDURE proc_GetRecurrenceSeriesData (  
    @ListID                uniqueidentifier,  
    @ItemID                int,  
    @EventType             int,  
    @EventTypeColName      nvarchar(64),  
    @EventTypeRowOrd       tinyint = 0,  
    @UIDColName            nvarchar(64),  
    @UIDRowOrd             tinyint = 0,  
    @StartDateColName     nvarchar(64),  
    @StartDateRowOrd      tinyint = 0,
```

```

    @MasterIDColName          nvarchar(64),
    @MasterIDRowOrd          tinyint = 0,
    @SeriesUID                uniqueidentifier OUTPUT,
    @StartDate                datetime OUTPUT,
    @MasterSeriesID          int OUTPUT,
    @RequestGuid              uniqueidentifier = NULL OUTPUT
);

```

**@ListId:** Specifies the identifier of a recurrence list.

**@ItemID:** Specifies the identifier of the meeting series item.

**@EventType:** Specifies the calendar event type to retrieve as defined by the meetings calendar event type (section [2.2.1.4](#)).

**@EventTypeColName:** Specifies the name of the column (1) in the **UserData** view of the content database that corresponds to the calendar event type. For more details about **UserData** view, see [\[MS-WSSFO2\]](#) section 2.2.7.8. MUST NOT be NULL.

**@EventTypeRowOrd:** The ordinal of list item that MUST contain the calendar event type information specified inside the column (1) identified by the parameter *@EventTypeColName*.

**@UIDColName:** The name of the column (1) in the **UserData** view of the content database that corresponds to the UID of the Instance.

**@UIDRowOrd:** The name of the row (3) ordinal in the **UserData** view of the content database that MUST contain the UID of the Instance in the column (1) specified by the *@UIDColName* parameter.

**@StartDateColName:** The name of the column (1) in the **UserData** view of the content database that corresponds to the start date of the recurring instance. MUST NOT be NULL.

**@StartDateRowOrd:** The ordinal of list item that MUST contain the start date information specified inside the column (1) identified by the parameter *@StartDateColName*.

**@MasterIDColName:** The name of the column (1) in the **UserData** view of the content database that corresponds to the master item identifier of the recurring instance. MUST NOT be NULL.

*@MasterIDRowOrd:* The ordinal of list item that MUST contain the master item identifier specified inside the column (1) identified by the parameter *@MasterIDColName*.

**@SeriesUID:** Specifies the meeting series identifier. This value is filled and returned to the caller.

**@StartDate:** Specifies the starting datetime for the series data. This value is filled and returned to the caller.

**@MasterSeriesID:** Specifies the master series identifier. This value is filled and returned to the caller. This value MUST be NULL if the specified meeting series item is not associated with a master item identifier in the **UserData** view.

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

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

Value	Description
"0"	Successful completion. <i>@SeriesUID</i> and <i>@StartDate</i> MUST NOT be NULL.

Value	Description
"4005"	No records found that match the input parameters. All output parameters are NULL.

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

### 3.1.5.25 proc\_GetResourcesAtScope

The **proc\_GetResourcesAtScope** stored procedure is called to get resource strings designated by *@WebId*, *@LCID* and *@OnlyUntranslated*. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetResourcesAtScope (
    @WebId                uniqueidentifier,
    @LCID                  int,
    @OnlyUntranslated     bit
);
```

**@WebId:** The identifier of the site (2).

**@LCID:** The LCID for the resource string.

**@OnlyUntranslated:** This parameter is ignored if *@LCID* is equal to the LCID of the site (2) specified by *@WebId*. If this parameter is "1", select localized resource strings, otherwise, select resource strings that are not translated.

If *@LCID* is equal to the LCID of the site (2) designated by *@WebId*, return the resource strings of the site (2) designated by *@WebId* of the locale designated by *@LCID*.

If *@LCID* is not equal to the LCID of the site (2) designated by *@WebId*, return the resource strings defined in the locale of the site (2) specified by *@WebId*, but not in the locale specified by *@LCID* and the following resource strings.

- If *@OnlyUntranslated* is "1", the localized resource strings of the locale designated by *@LCID* of the site (2) designated by *@WebId*.
- If *@OnlyUntranslated* is zero ("0"), the resource strings, which are not translated, of the locale designated by *@LCID* of the site (2) designated by *@WebId*.

**Return values:** MUST return an **integer** return code that MUST be zero.

**Result Sets:** MUST return one of the following result sets depending on the type of the resource.

Type of resource	Returned Result Set
Single line of text	One <b>GetResAtScopeNvar</b> result set
Multiple lines of text	One <b>GetResAtScopeNtext</b> result set

#### 3.1.5.25.1 Get Resources Scope Nvarchar Result Set

The **GetResAtScopeNvar** result set contains:

```
ListId                uniqueidentifier,
ResourceName          nvarchar(520),
BitType               bit,
```



```
NvarcharVal          nvarchar(256);
```

**ListId:** The identifier of the list (1).

**ResourceName:** The name of the resource string.

**BitType:** The type of the resource, as defined in section [2.2.1.10](#). This MUST be zero ("0") in this result set.

**NvarcharVal:** The resource value.

### 3.1.5.25.2 Get Resources Scope Ntext Result Set

The **GetResAtScopeNtext** result set contains

```
ListId                uniqueidentifier,  
ResourceName          nvarchar (520),  
BitType               bit,  
NtextVal              nvarchar(2^31-1);
```

**ListId:** The identifier of the list (1).

**ResourceName:** The name of the resource.

**BitType:** The type of the resource, as defined in section [2.2.1.10](#). This MUST be "1" in this result set.

**NtextVal:** The resource value.

### 3.1.5.26 proc\_GetVersionIndependentMetaInfo

The **proc\_GetVersionIndependentMetaInfo stored procedure** is invoked to get the version-independent metadata of a document. **proc\_GetVersionIndependentMetaInfo** is defined using T-SQL syntax, as follows:

```
PROCEDURE proc_GetVersionIndependentMetaInfo (  
    @SiteId            uniqueidentifier,  
    @DirName           nvarchar(256),  
    @LeafName          nvarchar(128),  
    @RequestGuid       uniqueidentifier = NULL OUTPUT  
);
```

**@SiteId:** The site collection identifier of the site collection containing the requested document.

**@DirName:** The directory name of the requested document.

**@LeafName:** The leaf name of the requested document.

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

**Return Values:** The stored procedure returns an **integer** that MUST be ignored.

**Result Sets:** MUST return one result set as defined in the following subsection.

### 3.1.5.26.1 VersionMetaInfo Result Set

The **VersionMetaInfo** result set returns the metadata and the metadata version of a document. The **VersionMetaInfo** result set MUST contain one row (3). The **VersionMetaInfo** result set is defined using T-SQL syntax, as follows:

```
UnVersionedMetaInfo      varbinary(max),
UnVersionedMetaInfoVersion  int;
```

**UnVersionedMetaInfo:** A metadict holding all version-independent metadata for the document.

**UnVersionedMetaInfoVersion:** An **integer** value that specifies the version of the **UnVersionedMetaInfo** metadata.

### 3.1.5.27 proc\_GetWebAndChildrenNSInfo

The **proc\_GetWebAndChildrenNSInfo** stored procedure is called to retrieve a set of properties as described in the result set section related to a site (2) and its subsites, document libraries and lists (1). The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetWebAndChildrenNSInfo(
    @SiteId          uniqueidentifier,
    @WebId           uniqueidentifier,
    @GetThisWeb      bit = 0,
    @GetSubwebs     bit = 0,
    @GetDocLibs     bit = 0,
    @GetLists       bit = 0,
    @Lcid           bit = 0,
    @RequestGuid    uniqueidentifier = NULL OUTPUT
);
```

**@SiteId:** The identifier of the site collection containing the site (2) to retrieve.

**@WebId:** The identifier of the site (2) to retrieve.

**@GetThisWeb:** If this parameter is "1", a **SiteNSInfo** result set of the site (2) specified by **@WebId** is returned. The default value is zero.

**@GetSubwebs:** If this parameter is "1", a **SiteNSInfo** result set of the subsites under the site (2) specified by **@WebId** is returned. The default value is zero.

**@GetDocLibs:** If this parameter is "1", a **ListNSInfo** result set of document library of the site (2) specified by **@WebId** is returned. If **@GetLists** is "1", the rows (3) in this result set are merged with the rows (3) in the result set of lists (1). The default value is zero.

**@GetLists:** If this parameter is "1", a **ListNSInfo** result set that contains the lists (1) except document libraries under the site (2) specified by **@SiteId** and **@WebId** is returned. If **@GetDocLibs** is "1", the rows (3) in this result set are merged with the rows (3) in the result set of document library. The default value is zero.

**@Lcid:** The LCID for the site (2) specified by **@WebId**.

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

**Return Values:** Returns an **integer** return code that MUST be zero.

**Result Sets:** MUST return zero to three result sets based on the values of the four bit flag input parameters: *@GetThisWeb*, *@GetSubWeb*, *@GetDocLibs* and *@GetLists*.

Input Parameter	Returned Result Set
<i>@GetThisWeb</i> = "1"	One <b>SiteNSInfo</b> result set.
<i>@GetSubWeb</i> = "1"	One <b>SiteNSInfo</b> result set.
<i>@GetDocLibs</i> = "1" OR <i>@GetLists</i> = "1"	One <b>ListNSInfo</b> result set.

### 3.1.5.27.1 SiteNSInfo Result Set

The **SiteNSInfo** result set returns a set of properties related to a specific site (2). The **SiteNSInfo** result set specified by *@GetThisWeb* MUST return one row (3) if the site (2) specified by *@SiteId* and *@WebId* can be found. It MUST return no rows (3) if the site (2) cannot be found. The **SiteNSInfo** result set specified by *@GetSubWebs* MUST return zero or multiple rows (3). The T-SQL syntax for the result set is as follows:

```

Title                nvarchar(255),
Id                   uniqueidentifier,
FullUrl              nvarchar(256),
WebTemplate          int,
ProvisionConfig      smallint,
MeetingCount        smallint,
UserResource         nvarchar(256) or nvarchar(max),
Acl                  image,
AnonymousPermMask   bigint;

```

**Title:** The title of the site (2).

**Id:** The identifier of the site (2).

**FullUrl:** The store-relative URL of the site (2).

**WebTemplate:** The identifier for the template used in the **site definition** to define the base structure of this site (2). The values are the same as defined by *ProvisionConfig*. See [\[MS-WSSFO2\]](#) section 2.2.5.22, for additional details.

**ProvisionConfig:** An identifier of the **site template** used to provision this site (2). The following reserved values are defined:

Value	Description
"-1"	This site (2) has no template provisioned.
"0"	This site (2) has the implementation-specific default template applied.
"1"	This site (2) has the Team Collaboration site template applied.
"2"	This site (2) has the Meeting Workspace site template applied.
"3"	This site (2) has the <b>Central Administration site</b> template applied.
"4"	This site (2) has the Wiki site template applied.

Value	Description
"9"	This site (2) has the Blog site template applied.

**MeetingCount:** If this subsite is a Meeting Workspace site, this value indicates the number of meetings that are configured. A value of "-1" indicates that the workspace contains a recurring meeting.

**UserResource:** The localized Web title specified by *@WebId* and *@Lcid*.

**Acl:** The binary serialization of the ACL for the site (2). The format is specified in [\[MS-WSSFO2\]](#) section 2.2.4.6.

**AnonymousPermMask:** A permissions mask indicating the rights granted to a user that is anonymous, or has no specific rights, to this site (2). The format is specified in [\[MS-WSSFO2\]](#) section 2.2.2.14.

### 3.1.5.27.2 ListNSInfo Result Set

The **ListNSInfo** result set returns a set of properties related to all lists (1) if *@GetLists* is "1" or document libraries if *@GetDocLibs* is "1" under the specified site (2) whose sort behavior MUST be like the sort behavior of folders. The **ListNSInfo** result set MUST contain zero or more rows (3). The T-SQL syntax for the result set is as follows:

```

tp_Title          nvarchar(255),
tp_Id             uniqueidentifier,
{FullUrl}         nvarchar(260),
tp_RootFolder     uniqueidentifier,
tp_ImageUrl       nvarchar(255),
tp_BaseType       int,
tp_ServerTemplate int,
FolderChildCount int,
UserResource      nvarchar(256) or nvarchar(max),
Acl               image,
AnonymousPermMask bigint;

```

**tp\_Title:** The title of the list (1) or document library.

**tp\_Id:** The list identifier of the list (1) or document library.

**{FullUrl}:** The store-relative URL of the list (1) or document library.

**tp\_RootFolder:** The identifier of the root folder for the list (1) or document library.

**tp\_ImageUrl:** The URL of the image used to represent the list (1) or document library.

**tp\_BaseType:** The base type of the list (1) or document library, as defined in [\[MS-WSSFO2\]](#) section 2.2.3.11.

**tp\_ServerTemplate:** The **List Server Template** enumeration value of the list template that defines the base structure of the list (1) or document library. See [\[MS-WSSFO2\]](#) section 2.2.3.12, for more details.

**FolderChildCount:** The number of child objects (1) of the root folder of the list (1) or document library.

**UserResource:** The localized titles of the lists (1) under the Web specified by @WebId and @Lcid.

**Acl:** The binary serialization of the ACL for the list (1) or document library. The format is specified in [\[MS-WSSFO2\]](#) section 2.2.4.6.

**AnonymousPermMask:** The permissions mask that applies to an anonymous user of the list (1) or document library. The format is specified in [\[MS-WSSFO2\]](#) section 2.2.2.14.

### 3.1.5.28 proc\_GetWebComments

The **proc\_GetWebComments** stored procedure is called to retrieve all Web discussion comments for a site (2). The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetWebComments (  
    @SiteId          uniqueidentifier,  
    @WebId           uniqueidentifier,  
    @RequestGuid     uniqueidentifier = NULL OUTPUT  
);
```

**@SiteId:** The site collection identifier for the site collection containing the specified site (2).

**@WebId:** The site identifier for the site (2) containing the requested Web discussion comments.

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

**Return Values:** Returns an **integer** return code that MUST be in the following table:

Value	Description
"0"	Successful completion.
"3"	Either @SiteId or @WebId is NULL.

**Result Sets:** MUST return the **GetWebComments** result set.

#### 3.1.5.28.1 GetWebComments Result Set

The **GetWebComments** result set returns all Web discussion comments for the specified site (2). This result set MUST be returned, and contains one row (3) for each Web discussion comment. The T-SQL syntax for the result set is as follows:

```
{DocFullUrl}    nvarchar(260),  
Id              int,  
Parent         int,  
CommentId     nvarchar(255),  
Bookmark      nvarchar(127),  
Author        nvarchar(255),  
UserId        int,  
Subject       nvarchar(255),  
Created       datetime,  
Status        smallint,  
Comment       ntext;
```

**{DocFullUrl}:** The store-relative URL of the document that this Web discussion comment is associated with.

**Id:** The identifier of the Web discussion comment.

**Parent:** The identifier of the parent Web discussion comment, if present. If the parent Web discussion comment does not exist, this MUST be zero.

**CommentId:** A protocol client specified identifier for the Web discussion comment.

**Bookmark:** A protocol client specified reference to the place in the document that the Web discussion comment refers to.

**Author:** A protocol client specified name for the user that added the Web discussion comment.

**UserId:** The user identifier for the user that added the Web discussion comment.

**Subject:** The subject of the Web discussion comment.

**Created:** The time in UTC when the Web discussion comment was created.

**Status:** A Web discussion comment status flag (section [2.2.2.3](#)) that indicates the status of a Web discussion comment.

**Comment:** The body text of the Web discussion comment.

### 3.1.5.29 `proc_GetWebNavAcls`

The `proc_GetWebNavAcls` stored procedure is called to fetch ACLs for all the unique security scopes of documents in the navigation structure. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetWebNavAcls(  
    @SiteId          uniqueidentifier,  
    @WebId           uniqueidentifier,  
    @NavParentWebId uniqueidentifier,  
    @Inherited       bit,  
    @RequestGuid     uniqueidentifier = NULL OUTPUT  
);
```

**@SiteId:** The site collection identifier of the site collection that MUST contain the sites (2) specified by `@WebId` and `@NavParentWebId`.

**@WebId:** The site identifier of the site (2) for which ACLs are requested.

**@NavParentWebId:** The site identifier of the navigational parent site of the site (2) specified by `@WebId`. This field (3) can be NULL. If it is not NULL, additional rows (3) MUST be included in the **Nav ACLs** result set (section [2.2.4.2](#)).

**@Inherited:** A bit specifying whether navigation structure from the site (2) is inherited by its subsites.

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

**Return Values:** Returns an **integer** return code that MUST be ignored.

**Result Sets:** MUST return one result set as defined in section [2.2.4.2](#).

### 3.1.5.30 proc\_GetWebNavStruct

The **proc\_GetWebNavStruct** stored procedure is called to return the navigation structure for a given site (2). The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetWebNavStruct(  
    @SiteId          uniqueidentifier,  
    @WebId           uniqueidentifier,  
    @Inherited       bit,  
    @RequestGuid     uniqueidentifier = NULL OUTPUT  
);
```

**@SiteId:** The site collection identifier for the site collection that contains the site (2) specified by *@WebId*.

**@WebId:** The identifier of the site (2) whose navigation structure has been requested.

**@Inherited:** A bit specifying whether the navigation structure from the site (2) is inherited by its subsites. If this parameter is "1", the **Site Scope** and **CachedDataVersion** result set MUST be returned.

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

**Return Values:** Returns an **integer** return code that MUST be ignored.

**Result Sets:** MUST return a **Nav ACLs** result set and a **Nav Data** result set. It MUST also return a **Site Scope** and **CachedDataVersion** result set as the first result set if *@Inherited* is set to "1".

#### 3.1.5.30.1 Site Scope and CachedDataVersion Result Set

Returns the security scope and **CachedDataVersion** of the site (2). If *@Inherited* is "1", this result set MUST be returned. Otherwise, this result set MUST NOT be returned. If the result set returns, it MUST return one row (3). The T-SQL syntax for the result set is as follows:

```
    CachedDataVersion    int,  
    ScopeId              uniqueidentifier;
```

**CachedDataVersion:** An **integer** specifying the version of the site's cached information.

**ScopeId:** The unique identifier for the security scope of the site (2).

#### 3.1.5.30.2 Nav ACLs Result Set

The **Nav ACLs** result set returns the ACLs of the documents represented by navigation nodes in the navigation structure of the site (2). This result set MUST return zero or more rows (3), as defined in section [2.2.4.2](#), where *@NavParentWebId* is NULL and the *@Inherited* value is the value of the *@Inherited* parameter.

#### 3.1.5.30.3 Nav Data Result Set

The **Nav Data** result set returns the information about the navigation structure of the site (2). It MUST return one or more rows (3). Note that this result set includes more information than the **Nav Data** result set specified in section [2.2.4.3](#). The T-SQL syntax for the result set is as follows:

```

Eid                int,
EidParent          int,
ElementType        tinyint,
{Url}              nvarchar(260),
Name               nvarchar(256),
NodeMetainfo       varbinary(max),
NonNavPage         bit,
NavSequence        bit,
ChildOfSequence    bit,
DateLastModified   datetime,
{ScopeId}          uniqueidentifier,
{SecurityType}     int,
NameResource        nvarchar(256);
Type               tinyint;

```

**Eid:** The navigation node element identifier of the navigation node.

**EidParent:** The navigation node element identifier of the parent navigation node.

**ElementType:** The navigation node type (section [2.2.1.5](#)) of the navigation node.

**{Url}:** The URL to which the navigation node points. This value MUST NOT be NULL if the **Eid** value is not zero.

**Name:** The display name of the navigation node.

**NodeMetainfo:** A binary serialization of the metadata for a navigation node in metadict form ([\[MS-FPSE\]](#) section 2.2.2.2.11). This MAY be NULL.

**NonNavPage:** A bit specifying whether the navigation node is hidden when rendering the navigation structure. If the bit is "1", the navigation node SHOULD be omitted [<5>](#). If it is set to zero ("0"), it MUST NOT be filtered out.

**NavSequence:** A bit specifying whether the navigation node represents a link bar. If this navigation node represents a link bar, this bit MUST be "1", else it MUST be zero.

**ChildOfSequence:** A bit specifying whether the navigation node is a child of a navigation node that represents a link bar. If it does, this bit MUST be "1". Otherwise, it MUST be zero.

**DateLastModified:** The time in UTC when the navigation node information was last modified.

**{ScopeId}:** The **scope identifier** of the document that the navigation node points to. If the navigation node does not point to a document, this MUST be "0x00" (and can be cast as uniqueidentifier, value "00000000-0000-0000-0000-000000000000").

**{SecurityType}:** Specifies which permissions are required for the user to be able to see this navigation node. The following table lists all possible valid values:

Value	Description
"0"	This navigation node is always visible to the user.
"1"	This navigation node requires the user to have permissions to view Web pages in the site (2) that are not in a list (1) or document library.
"2"	This navigation node requires the user to have permissions to view Web pages in the site (2) that are form (1) or list view pages of a list (1) or document library.



Value	Description
"3"	This navigation node requires the user to have permissions to view Web pages in the site (2) that are items in a document library.

**NameResource:** The resource name for the navigation node.

**Type:** If the navigation node does not point to a document, this MUST be NULL. Otherwise, this MUST be the **document store type** of the document that the navigation node points to. Refer to [\[MS-WSSFO2\]](#), section [2.2.2.4](#).

### 3.1.5.31 `proc_GetWebNavStructNodeByIds`

The **proc\_GetWebNavStructNodeByIds** stored procedure is called to fetch information about a single navigation node in a site (2). The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetWebNavStructNodeByIds (
    @SiteId          uniqueidentifier,
    @WebId           uniqueidentifier,
    @Eid            int,
    @RequestGuid    uniqueidentifier = NULL OUTPUT
);
```

**@SiteId:** The site collection identifier for the site collection that contains the site (2) specified by *@WebId*.

**@WebId:** The identifier of the site (2) that contains the navigation node.

**@Eid:** An **integer** that specifies the navigation node element identifier of the navigation node for which information is requested.

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

**Return Values:** Returns an **integer** return code that MUST be zero.

**Result Sets:** MUST return 2 result sets as follows: **Nav Data** result set (section [2.2.4.3](#)) and **Nav Children** result set (section [2.2.4.4](#)), respectively.

### 3.1.5.32 `proc_GetWebSubscriptions`

The **proc\_GetWebSubscriptions** stored procedure is called to return alert subscriptions for a given site collection, site (2) and user identifier. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetWebSubscriptions (
    @SiteId          uniqueidentifier,
    @WebId           uniqueidentifier,
    @UserId          int,
    @RequestGuid    uniqueidentifier = NULL OUTPUT
);
```

**@SiteId:** The site collection identifier of the site collection for which the alert subscriptions are requested to be returned. If *@SiteId* is NULL then this stored procedure returns an empty result set.

**@WebId:** The identifier of the site (2) for which the alert subscriptions are requested to be returned. If *@WebId* is NULL then this stored procedure returns an empty result set.

**@UserId:** The user identifier for the user for which the alert subscriptions are requested to be returned.

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

**Return Values:** Returns an **integer** that MUST be zero.

**Result Sets:** MUST return one result set as described in the following subsections.

### 3.1.5.32.1 WebSubscriptions Result Set

The **WebSubscriptions** result set contains the alert subscriptions for the site collection specified by *@SiteId*, the site (2) specified by *@WebId*, and the user identifier specified by *@UserId*. If *@SiteId* is NULL or if *@WebId* is NULL, this stored procedure returns an empty result set. If *@UserId* is NULL, alert subscriptions are not filtered by the **UserId** and the result set contains all alert subscriptions for all users. The **WebSubscriptions** result set MUST contain zero rows (3) if no matching alert subscriptions are found. Otherwise it MUST contain one row (3) for each matching alert subscription. The T-SQL syntax for the result set is as follows:

Id	uniqueidentifier,
ListId	uniqueidentifier,
ItemId	int,
EventType	int,
NotifyFrequency	int,
NotifyTime	datetime,
Status	tinyint,
UserId	int,
ListUrl	nvarchar(256),
ListTitle	nvarchar(255),
ItemName	nvarchar(255),
AlertTitle	nvarchar(1000),
AlertType	int,
AlertTemplateName	nvarchar(255),
Filter	nvarchar(4000),
BinaryFilter	varbinary(1024),
Properties	ntext,
DeliveryChannel	int;

**Id:** The identifier of this alert subscription. The value MUST NOT be NULL.

**ListId:** The list identifier with which this alert subscription is associated. The value MUST NOT be NULL.

**ItemId:** The item identifier with which this alert subscription is associated. This MUST NOT be NULL if the alert (1) type (section [2.2.1.3](#)) of the alert subscription is Item alert and MUST be NULL if the alert (1) type of the alert subscription is List alert.

**EventType:** An alert (1) event (2) type (section [2.2.2.1](#)) that specifies an **integer** mask for the types of events (2) on which to send the notification. This value MUST NOT be NULL.

**NotifyFrequency:** A notification frequency type (section [2.2.1.6](#)) that specifies an **integer** indicating the frequency of the alert (1) notification. This value MUST NOT be NULL.

**NotifyTime:** The time for which the alert (1) is scheduled. This value MUST NOT be NULL for scheduled alert subscriptions for a given site collection, site (2), and user. It MUST be NULL for immediate alert subscriptions.

**Status:** An alert (1) status type (section [2.2.1.2](#)) that specifies an **integer** indicating the status of the alert subscription. This value **MUST NOT** be NULL.

**UserId:** The user identifier of the user for whom the alert (1) has been created. This value **MUST NOT** be NULL.

**ListUrl:** The URL of the list (1) with which this alert subscription is associated. The value **MUST NOT** be NULL.

**ListTitle:** The title of the list (1) with which this alert subscription is associated.

**ItemName:** The name of the list item associated with the alert subscription. This value **MUST** be NULL for the following three cases:

- **ItemId** is NULL.
- The alert subscription is for an item in a **links list**.
- The base type of the list (1) is "issue" ([\[MS-WSSFO2\]](#) section 2.2.3.11).

**AlertTitle:** The title of the alert (1) of the alert subscription.

**AlertType:** An **integer** whose lower 8 bits specifies the type of the alert (1), as specified in section [2.2.1.3](#). This **MUST** also contain the special alert (1) flags (section [2.2.2.2](#)) for an always notify alert or a system alert.

**AlertTemplateName:** The name of the alert template of the alert subscription.

**Filter:** The syntax query in CAML for a filter to apply to the alert (1) associated with this alert subscription.

**BinaryFilter:** The compiled binary syntax query in CAML for a filter to apply to the alert (1) associated with this alert subscription.

**Properties:** An XML blob representing the properties of the alert subscription.

**DeliveryChannel:** Specifies the delivery channel for the alert subscription, as defined in section [2.2.1.1](#).

### 3.1.5.33 **proc\_GetWebSubscriptionsForBackup**

The **proc\_GetWebSubscriptionsForBackup** stored procedure is called to return alert subscriptions associated with a site collection and a site (2). The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetWebSubscriptionsForBackup(  
    @SiteId          uniqueidentifier,  
    @WebId           uniqueidentifier,  
    @RequestGuid    uniqueidentifier = NULL OUTPUT  
);
```

**@SiteId:** The site collection identifier for the site collection for which the alert subscriptions are requested to be returned.

**@WebId:** The identifier of the site (2) for which the alert subscriptions are requested to be returned.

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

**Return Values:** Returns an **integer** that MUST be zero.

**Result Sets:** MUST return a single result set as defined in the following subsections.

### 3.1.5.33.1 WebSubscriptionsForBackupResult Set

The **WebSubscriptionsForBackup** result set contains the alert subscriptions, and their associated document information, for the site collection specified in the *@SiteId* parameter and the site (2) specified in the *@WebId* parameter. The **WebSubscriptionsForBackup** result set always returns and MUST contain zero rows (3) if there are no current alert subscriptions for the site collection specified in the *@SiteId* parameter and the site (2) specified in the *@WebId* parameter; otherwise, if there are current alert subscriptions, the **WebSubscriptionsForBackup** result set MUST contain the number of rows (3) equal to the number of current alert subscriptions. The T-SQL syntax for the result set is as follows:

```
Id                uniqueidentifier,  
ListId            uniqueidentifier,  
ItemId            int,  
EventType         int,  
NotifyFrequency  int,  
UserId            int,  
UserEmail         nvarchar(255),  
DirName           nvarchar(256),  
LeafName          nvarchar(128);
```

**Id:** The identifier of the alert subscription. The value MUST NOT be NULL.

**ListId:** The list identifier of the item referenced by the alert subscription. The value MUST NOT be NULL.

**ItemId:** The item identifier of the item referenced by the alert subscription. This MUST NOT be NULL if the alert (1) type (section [2.2.1.3](#)) of the alert subscription is Item alert and MUST be NULL if the alert (1) type of the alert subscription is List alert.

**EventType:** An alert (1) event (2) type (section [2.2.2.1](#)) that specifies an **integer** mask for the types of events (2) on which to send the notification. This value MUST NOT be NULL.

**NotifyFrequency:** A notification frequency type (section [2.2.1.6](#)) that specifies an **integer** indicating the frequency of the alert (1) notification. This value MUST NOT be NULL.

**UserId:** The user identifier of the user for whom the alert (1) has been created. The value MUST NOT be NULL.

**UserEmail:** The e-mail address of the user. The value MUST NOT be NULL.

**DirName:** The directory name of the document referred to by the alert subscription. The value MUST be NULL if the alert subscription does not refer to a document.

**LeafName:** The leaf name, or file name, of the document referred to by the alert subscription. The value MUST be NULL if the alert subscription does not refer to a document.

### 3.1.5.34 proc\_GetWebSubscriptionsUniqueUsers

The **proc\_GetWebSubscriptionsUniqueUsers** stored procedure is called to return the login name and display name of all users with active alert subscriptions. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetWebSubscriptionsUniqueUsers(  
    @SiteId          uniqueidentifier,  
    @WebId           uniqueidentifier,  
    @RequestGuid     uniqueidentifier = NULL OUTPUT  
);
```

**@SiteId:** The site collection identifier of the site collection. @SiteId MUST NOT be NULL.

**@WebId:** The site identifier of the site (2). @WebId MUST NOT be NULL.

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

**Return Values:** Returns an **integer** value that MUST be zero.

**Result Sets:** MUST return a single result set as defined in the following subsection.

#### 3.1.5.34.1 WebSubscriptionsUniqueUsers Result Set

WebSubscriptionsUniqueUsers contains a list of login name and display name for all users who have nondeleted alert subscriptions for the site collection specified in the @SiteId parameter and the site (2) specified in the @WebId parameter.

The **WebSubscriptionsUniqueUsers** result set MUST contain zero rows (3) if there are no nondeleted alert subscriptions for the site collection specified in the @SiteId parameter and the site (2) specified in the @WebId parameter. Otherwise if there are nondeleted alert subscriptions, the **WebSubscriptionsUniqueUsers** result set MUST contain the number of distinct users for whom there are nondeleted alert subscriptions. The **WebSubscriptionsUniqueUsers** result set MUST be sorted by **tp\_Login**.

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

```
tp_Login          nvarchar(255);  
tp_Title          nvarchar(255);
```

**tp\_Login:** The login name of a user with at least one nondeleted alert subscription. This value MUST NOT be NULL.

**tp\_Title:** The display name of a user with at least one nondeleted alert subscription. This value MUST NOT be NULL.

### 3.1.5.35 proc\_HandleMtgRecurPatternChange

The **proc\_HandleMtgRecurPatternChange** stored procedure is called to update the calendar events in a meeting series when the overall recurrence pattern has changed. Specifically, 3 updates are made to the existing meeting calendar events:

1. Any delete exceptions in the range [*@InstanceIDStart*, *@InstanceIDEndDel*] are removed.

2. Any modify exceptions in the range [*@InstanceIDStart*, *@InstanceIDEndSplit*] are split from the series and become standalone calendar events.
3. Any future modify exceptions after *@InstanceIDEndSplit* are orphaned.

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

```

PROCEDURE proc_HandleMtgRecurPatternChange (
    @SiteId                uniqueidentifier,
    @WebId                 uniqueidentifier,
    @MtgListID            uniqueidentifier,
    @InstanceIDStart      int,
    @InstanceIDEndDel     int,
    @InstanceIDEndSplit   int,
    @DTStamp              datetime,
    @UserId               int,
    @UserTitle            nvarchar(255),
    @InstanceIDSplitMax   int = NULL OUTPUT,
    @RequestGuid          uniqueidentifier = NULL OUTPUT
);

```

**@SiteId:** Specifies the site collection identifier for the site collection in which the meeting series exists. *@SiteId* MUST NOT be NULL.

**@WebId:** The identifier of the site (2) in which the meeting series exists. *@WebId* MUST NOT be NULL.

**@MtgListID:** Specifies the identifier of the list (1) representing the meeting series data.

**@InstanceIDStart:** Specifies the starting instance to modify.

**@InstanceIDEndDel:** Specifies the ending instance for removing delete exceptions.

**@InstanceIDEndSplit:** Specifies the ending instance for splitting modify exceptions, all modify exceptions after this one will be orphaned.

**@DTStamp:** Specifies the datetime stamp for the meeting series. Modify exceptions before this date are split off from the series and made standalone calendar events. Modify exceptions after this date are orphaned.

**@UserId:** Specifies the user identifier of the meeting organizer.

**@UserTitle:** Specifies the **user name** of the meeting organizer.

**@InstanceIDSplitMax:** Returns the highest instance identifier that is now split off from the series. This value is filled and returned to the caller.

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

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

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

### 3.1.5.36 proc\_HTCreatRow

The **proc\_HTCreatRow** stored procedure is called to put a file into the HTML translate cache [<6>](#). The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_HTCreateRow(
    @DocSiteId      uniqueidentifier,
    @DocDirName     nvarchar(256),
    @DocLeafName    nvarchar(128),
    @TransName      nvarchar(128),
    @JobType        tinyint,
    @File           varbinary(max),
    @MainFile       bit,
    @maxCacheSize   int,
    @RequestGuid    uniqueidentifier = NULL OUTPUT
);

```

**@DocSiteId:** The site collection identifier of the site collection containing the original document.

**@DocDirName:** Directory name for the original document.

**@DocLeafName:** Leaf name for the original document.

**@TransName:** The name of the translated file being put into the HTML translate cache.

**@JobType:** A value specifying the browser type that this HTML translate cache file was created for. This value **MUST** be one of the values listed in the following table:

Value	Description
"0"	The raw user agent string of the client browser starts with "Mozilla/1", "Mozilla/3" or "Mozilla/4".
"1"	All other values of the raw user agent string of the client browser.
"2"	HTML Transform of document failed or was not allowed.

**@File:** The content of the file being stored in the HTML translate cache.

**@MainFile:** Specifies whether or not this is the main file for the set of translated files associated with the original document. This value **MUST NOT** be NULL.

**@maxCacheSize:** The maximum total size allowed for all HTML transformed files associated with the original document for this *@JobType*. This value is specified in megabytes.

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

**Return Values:** **MUST** return an **integer** return code that **MUST** be in the following table:

Value	Description
"0"	Successful completion.
"-4"	Cache entry already exists with the same <i>@DocSiteId</i> , <i>@DocDirName</i> , <i>@DocLeafName</i> and <i>@TransName</i> .

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

### 3.1.5.37 proc\_HTGetFile

The **proc\_HTGetFile** stored procedure is called to get the contents of a single file from the HTML translate cache. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_HTGetFile(
    @DocSiteId      uniqueidentifier,
    @DocDirName     nvarchar(256),
    @DocLeafName    nvarchar(128),
    @TransName      nvarchar(128),
    @JobType        tinyint,
    @RequestGuid    uniqueidentifier= NULL OUTPUT
);

```

**@DocSiteId:** The site collection identifier of the site collection containing the original document.

**@DocDirName:** Directory name for the original document.

**@DocLeafName:** Leaf name for the original document.

**@TransName:** The name of the translated file being retrieved from the HTML translate cache.

**@JobType:** A value specifying the browser type that this HTML translate cache file was created for. This MUST be one of the values in the following table:

Value	Description
"0"	The raw user agent string of the client browser starts with "Mozilla/1", "Mozilla/3" or "Mozilla/4".
"1"	All other values of the raw user agent string of the client browser.
"2"	HTML Transform of document failed or was not allowed.

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

**Return Values:** MUST return an **integer** return code of zero.

**Result Sets:** MUST return the **File** result set.

### 3.1.5.37.1 File Result Set

The **File** result set returns the file that uniquely matches the input parameters. This result set MUST return one row (3) containing the file, if a match for the file is found. Otherwise, it MUST return zero rows (3). The T-SQL syntax for the result set is as follows:

```
{File}          image;
```

**{File}:** The file's content.

### 3.1.5.38 proc\_HTGuidFromOrig

The **proc\_HTGuidFromOrig** stored procedure is called to get the name of the main file for the set of translated files in the HTML translate cache associated with the original document. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_HTGuidFromOrig(
    @DocSiteId      uniqueidentifier,
    @DocDirName     nvarchar(256),
    @DocLeafName    nvarchar(128),
    @JobType        tinyint,

```



```

        @RequestGuid      uniqueidentifier = NULL OUTPUT
    );

```

**@DocSiteId:** The site collection identifier of the site collection containing the original document.

**@DocDirName:** Directory name for the original document.

**@DocLeafName:** Leaf name for the original document.

**@JobType:** A value specifying the browser type that this HTML translate cache file was created for. This value **MUST** be one of the values in the following table:

Value	Description
"0"	The raw user agent string of the client browser starts with "Mozilla/1", "Mozilla/3" or "Mozilla/4".
"1"	All other values of the raw user agent string of the client browser.
"2"	HTML Transform of document failed or was not allowed.

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

**Return Values:** Returns an **integer** return code that **MUST** be in the following table:

Value	Description
"0"	Successful completion.
"5"	Access is denied.

**Result Sets:** **MUST** return a **Translation Name** result set.

### 3.1.5.38.1 Translation Name Result Set

The **Translation Name** result set returns the name of the main file for the set of translated files in the HTML translate cache associated with the original document. This result set **MUST** contain one row (3) if a corresponding row exists. The client **MUST** use the data in the first row (3) and ignore the rest. If none exist, it **MUST** return zero rows (3). The T-SQL syntax for the result set is as follows:

```

TransName          nvarchar(128);

```

**TransName:** The name of the main file for the set of translated files in the HTML translate cache associated with the original document.

### 3.1.5.39 proc\_InstantiateMtgSeriesOccurrence

The **proc\_InstantiateMtgSeriesOccurrence** stored procedure is called to instantiate an occurrence of a recurring meeting series. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_InstantiateMtgSeriesOccurrence(
    @SiteId          uniqueidentifier,
    @WebId           uniqueidentifier,
    @InstanceID     int,

```

```

@DTStartUTC          datetime,
@CreateOrphaned      bit = 0,
@ForAttendees        bit = NULL,
@AlreadyInstantiated bit = NULL OUTPUT,
@RequestGuid         uniqueidentifier = NULL OUTPUT
);

```

**@SiteId:** Specifies the site collection identifier for the site collection in which the meeting series exists. *@SiteId* MUST NOT be NULL.

**@WebId:** The identifier of the site (2) in which the meeting series exists. *@WebId* MUST NOT be NULL.

**@InstanceID:** Specifies the identifier of the meeting instance containing the occurrence to instantiate. *@InstanceID* MUST NOT be NULL.

**@DTStartUTC:** Specifies the start date of the meeting series in UTC.

**@CreateOrphaned:** Specifies whether the occurrence should be orphaned from the meeting series. Zero ("0"), which is the default, means do not orphan the occurrence. "1" means orphan the occurrence.

**@ForAttendees:** Specifies whether to copy over attendee series data as well to the new occurrence. "1" means copy only attendees, zero ("0") means copy all except attendees, and NULL, which is the default, mean copy all.

**@AlreadyInstantiated:** Specifies the output parameter that is set to "1" if the meeting was already instantiated. Otherwise, it is set to zero ("0").

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

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

Value	Description
"0"	Successful completion. The meeting series occurrence was successfully instantiated.
"3"	The target meeting series item corresponding to the specified <i>@SiteId</i> and <i>@WebId</i> could not be created.
"87"	Invalid parameter encountered when creating the meeting series item. <i>@SiteId</i> or <i>@WebId</i> contains invalid data.
"13"	Two possible failure conditions: <ul style="list-style-type: none"> <li>▪ No meeting series found for the given <i>@WebId</i></li> <li>▪ List (1) name not specified in default meeting instance</li> </ul>
"80"	The meeting occurrence was already instantiated.
"160"	Bad argument used to create the meeting series item. <i>@SiteId</i> or <i>@WebId</i> contains invalid data.
"183"	An instance for this meeting already exists where the calendar event type is not equal to "2".
"212"	Site collection locked. The operation could not be performed because the site collection containing the meeting series item is in <b>read-only mode</b> .

Value	Description
"1150"	Unable to create the new occurrence; a newer version of Windows is required.
"1816"	Disk quota error. The quota for the site collection has reached the maximum allowable limit.

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

### 3.1.5.40 **proc\_ListDocsHavingComments**

The **proc\_ListDocsHavingComments** stored procedure is called to get the collection of documents in a given folder or site (2) that has at least one Web discussion comment. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_ListDocsHavingComments (
    @SiteId          uniqueidentifier,
    @WebUrl          nvarchar(256),
    @DocDir          nvarchar(256),
    @Collation       nvarchar(32),
    @RequestGuid     uniqueidentifier = NULL OUTPUT
);

```

**@SiteId:** The identifier of the site collection containing the documents whose Web discussion comments are to be retrieved. @SiteId MUST NOT be NULL.

**@WebUrl:** The store-relative URL to the site (2) containing the documents whose Web discussion comments are to be retrieved.

**@DocDir:** The store-relative URL to the folder containing the documents whose Web discussion comments are to be retrieved. If this parameter is NULL, all documents with at least one Web discussion comment in the site (2) MUST be returned.

**@Collation:** A Windows collation name string identifier that follows the format for the T-SQL COLLATE clause. This MUST be the collation name of one of the valid collation order values, with the case-insensitive and accent-sensitive flags set. For example, the default collation order is 25, Latin1\_General, which with the case-insensitive and accent-sensitive flags set has a SQL collation name string of "Latin1\_General\_CI\_AS". This MUST NOT be NULL.

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

**Return Values:** Returns an **integer** return code that MUST be in the following table:

Value	Description
"0"	Successful completion.
"3"	The site (2) identified by @SiteId and @WebUrl could not be found.

**Result Sets:** MUST return the **ListDocsHavingComments** result set when the return code is zero.

#### 3.1.5.40.1 **ListDocsHavingComments Result Set**

The **ListDocsHavingComments** result set returns a collection of documents in the specified location containing the Web discussion comments, along with the count of Web discussion

comments for each document. This result set contains one row (3) per document with at least one Web discussion comment. The T-SQL syntax for the result set is as follows:

```
Id                uniqueidentifier,  
DirName           nvarchar(256),  
LeafName          nvarchar(128),  
{Count}          int;
```

**Id:** The identifier of the document.

**DirName:** The directory name of the document.

**LeafName:** The leaf name of the document.

**{Count}:** The count of Web discussion comments associated with the document.

### 3.1.5.41 **proc\_MatchSchedSubscriptions**

The **proc\_MatchSchedSubscriptions** stored procedure is called to return scheduled alert subscriptions that are due and for which alert (1) notifications haven't been sent out, for a given site collection and notification frequency. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_MatchSchedSubscriptions(  
    @SiteId        uniqueidentifier,  
    @NotifyFreq    int,  
    @bAlwaysNotify bit,  
    @EventTime     datetime,  
    @RequestGuid   uniqueidentifier = NULL OUTPUT  
);
```

**@SiteId:** The site collection identifier of the site collection for which the alert subscriptions are requested to be returned. The value **MUST** not be NULL.

**@NotifyFreq:** A notification frequency type (section [2.2.1.6](#)) that specifies an **integer** indicating the frequency of the alert (1) notification.

**@bAlwaysNotify:** A bit indicating whether the event (2) information is returned along with the alert subscriptions that are requested to be returned. The value **MUST** be listed in the following table:

Value	Description
"0"	Return event (2) and alert subscription information.
"1"	Return only alert subscription information.

**@EventTime:** A datetime in UTC indicating the current time.

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

**Return Values:** **MUST** return an **integer** that **MUST** be zero.

**Result Sets:** **MUST** return a single result set as defined in the following subsection.

### 3.1.5.41.1 SchedSubscriptions Result Set

The **SchedSubscriptions** result set contains the alert subscriptions for the site collection specified by *@SiteId* and notify frequency specified by *@NotifyFreq*. The notify frequency is the interval at which alert (1) notifications are sent to the user. The **SchedSubscriptions** result set MUST be returned. The T-SQL syntax for the result set is as follows:

WebId	uniqueidentifier,
ListId	uniqueidentifier,
UserId	int,
UserEmail	nvarchar (255) ,
SiteUrl	nvarchar (136) ,
WebUrl	nvarchar (256) ,
ListUrl	nvarchar (256) ,
WebTitle	nvarchar (255) ,
WebLanguage	int,
WebLocale	int,
WebTimeZone	smallint,
WebTime24	bit,
WebCalendarType	smallint,
WebAdjustHijriDays	smallint,
ListTitle	nvarchar (255) ,
ListBaseType	int,
ListServerTemplate	int,
{ItemName}	nvarchar (255) ,
{ItemFullUrl}	nvarchar (260) ,
{ModifiedBy}	nvarchar (255) ,
{TimeLastModified}	datetime,
Id	uniqueidentifier,
{Id}	bigint,
EventType	int,
{EventType}	int,
{ItemId}	int,
NotifyTime	datetime,
NotifyFreq	int,
Properties	ntext,
AlertTitle	nvarchar (1000) ,
AlertType	int,
AlertTemplateName	nvarchar (255) ,
DeliveryChannel	int,
{EventData}	image,
{LookupFieldPermissionResults}	image;

**WebId:** The site identifier with which this alert subscription is associated. The value MUST NOT be NULL.

**ListId:** The list identifier with which this alert subscription is associated. The value MUST NOT be NULL.

**UserId:** The user identifier with which this alert subscription is associated. The value MUST NOT be NULL.

**UserEmail:** The e-mail address of the user with which this alert subscription is associated. The value MUST NOT be NULL.

**SiteUrl:** The URL of the site collection with which this alert subscription is associated. The value MUST NOT be NULL.

**WebUrl:** The URL of the site (2) with which this alert subscription is associated. The value MUST NOT be NULL.

**ListUrl:** The URL of the list (1) with which this alert subscription is associated. The value MUST NOT be NULL.

**WebTitle:** The title of the site (2) with which this alert subscription is associated.

**WebLanguage:** The language identifier of the site (2) with which this alert subscription is associated. The value MUST NOT be NULL.

**WebLocale:** The LCID of the site (2) with which this alert subscription is associated. The value MUST NOT be NULL.

**WebTimeZone:** The time zone of the site (2) with which this alert subscription is associated. The value MUST NOT be NULL.

**WebTime24:** A bit value that specifies whether the time displayed for the site (2) with which this alert subscription is associated is in 12-hour or 24-hour format. A value of zero ("0") indicates 12-hour format, and a value of "1" indicates 24-hour format. The value MUST NOT be NULL.

**WebCalendarType:** Contains calendar type, for non-Gregorian calendars, of the site (2) with which this alert subscription is associated. The value MUST NOT be NULL.

**WebAdjustHijriDays:** The number of days to extend or reduce the current month in Hijri calendars on the site (2) with which this alert subscription is associated. The value MUST NOT be NULL.

**ListTitle:** The title of the list (1) with which this alert subscription is associated. The value MUST NOT be NULL.

**ListBaseType:** The base type of the list (1) with which this alert subscription is associated. The value MUST NOT be NULL.

**ListServerTemplate:** The list template of the list (1) with which this alert subscription is associated. The value MUST NOT be NULL.

**{ItemName}:** A string that specifies the name of the object (1) under which this event (2) has occurred. The value MUST be NULL when *@bAlwaysNotify* is "1".

**{ItemFullUrl}:** A string that specifies the **server-relative URL** of the object (1) under which this event (2) has occurred. The value MUST be NULL when *@bAlwaysNotify* is "1".

**{ModifiedBy}:** A string that specifies the login name of a security principal (2) who added this event (2). The value MUST be NULL when *@bAlwaysNotify* is "1".

**{TimeLastModified}:** A Time Stamp in UTC that specifies the time when this event (2) was last modified. The value MUST be NULL when *@bAlwaysNotify* is "1".

**Id:** The identifier of this alert subscription. The value MUST NOT be NULL.

**{Id}:** The identifier of this event (2) in the change log. The value MUST be NULL when *@bAlwaysNotify* is "1".

**EventType:** Specified in section [2.2.2.1](#).

**{EventType}:** A 4-byte unsigned **integer** bit mask that specifies the type of an event (2). If the value is NOT NULL, it MUST have one or more of the flags that are specified in [\[MS-WSSDLIM\]](#) section 2.2.2.2. The value MUST be NULL when *@bAlwaysNotify* is "1".

**{ItemId}**: The **integer** that identifies a list item on which the event (2) has occurred. The value MUST be NULL when *@bAlwaysNotify* is "1".

**NotifyTime**: The time in UTC that the alert (1) is scheduled for.

**NotifyFreq**: A notification frequency type (section [2.2.1.6](#)) that specifies an **integer** indicating the frequency of the alert (1) notification. This value MUST NOT be NULL.

**Properties**: An XML blob representing the properties of the alert (1).

**AlertTitle**: The title of the alert (1) of the alert subscription.

**AlertType**: An **integer** whose lower eight bits specify the type of the alert (1), as specified in section [2.2.1.3](#). This MUST also contain the special alert (1) flags (section [2.2.2.2](#)) for an always notify alert or a system alert.

**AlertTemplateName**: The name of the alert template of the alert subscription.

**DeliveryChannel**: Specifies the delivery channel for the alert subscription, as defined in section [2.2.1.1](#).

**{EventData}**: Contains implementation-specific event (2) data significant to the front-end Web server but otherwise opaque to the back-end database server. This data is stored in the change log in the back-end database server along with this event (2). The value MUST be NULL when *@bAlwaysNotify* is "1".

**{LookupFieldPermissionResults}**: Contains implementation-specific data over the permissions the user has for **lookup fields**. The value MUST be NULL when *@bAlwaysNotify* is "1".

### 3.1.5.42 **proc\_ModifySubscription**

The **proc\_ModifySubscription** stored procedure is called to modify an existing alert subscription. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ModifySubscription(  
    @SiteId                uniqueidentifier,  
    @SubId                 uniqueidentifier,  
    @NotifyFreq           int,  
    @NotifyTime           datetime,  
    @NotifyTimeUTC        datetime,  
    @Status               tinyint,  
    @WebId                uniqueidentifier,  
    @EventType           int,  
    @UserId               int,  
    @SiteUrl              nvarchar(136),  
    @WebUrl               nvarchar(256),  
    @WebTitle             nvarchar(255),  
    @WebLanguage          int,  
    @WebLocale            int,  
    @WebTimeZone          smallint,  
    @WebTime24            bit,  
    @WebCalendarType      smallint,  
    @WebAdjustHijriDays   smallint,  
    @AlertTitle           nvarchar(1000),  
    @AlertType            int,  
    @AlertTemplateName   nvarchar(256),  
    @Filter               nvarchar(4000),  
    @BinaryFilter         varbinary(1024),
```

```

        @Properties                nvarchar(max),
        @DeliveryChannel           int,
        @UserEmail                 nvarchar(255) = NULL OUTPUT,
        @ItemName                  nvarchar(255) = NULL OUTPUT,
        @ItemId                     int OUTPUT,
        @RequestGuid                uniqueidentifier = NULL OUTPUT
    );

```

**@SiteId:** The site collection identifier for the site collection in which the list (1) or item exists. This value MUST NOT be NULL.

**@SubId:** The GUID of the alert subscription that is being modified. This value MUST NOT be NULL.

**@NotifyFreq:** A notification frequency type (section [2.2.1.6](#)) that specifies an **integer** indicating the frequency of the alert (1) notification.

**@NotifyTime:** The time, in local time for the site (2) in which the list (1) or item exists, at which the alert (1) is scheduled for. This value MUST NOT be NULL.

**@NotifyTimeUTC:** The time in UTC when the alert (1) is supposed to be sent. This value MUST NOT be NULL.

**@Status:** An alert (1) status type (section [2.2.1.2](#)) that specifies an **integer** indicating the status of the alert subscription. This value MUST NOT be NULL.

**@WebId:** The site identifier for the site (2) in which the list (1) or item exist. This value MUST NOT be NULL.

**@EventType:** An alert (1) event (2) type (section [2.2.2.1](#)) that specifies an **integer** mask for the types of events (2) on which to fire the notification. This value MUST NOT be NULL.

**@UserId:** The user identifier of the user for whom the alert (1) is modified. This value MUST NOT be NULL.

**@SiteUrl:** The URL of the site collection in which the list (1) or item exists. This value MUST NOT be NULL.

**@WebUrl:** The URL of the site (2) in which the list (1) or item exists. This value MUST NOT be NULL.

**@WebTitle:** The title of the site (2) in which the list (1) or item exists.

**@WebLanguage:** The LCID of the display language of the site (2) containing the list (1) or item. This value MUST NOT be NULL.

**@WebLocale:** An Integer representing the LCID of the site (2) locale. This value MUST NOT be NULL.

**@WebTimeZone:** The time zone identifier for the time zone to be used when displaying time values for this site (2). This value MUST NOT be NULL.

**@WebTime24:** A Boolean indicating if the Time is in a 24 hour format or a 12 hour format. This value MUST NOT be NULL.

**@WebCalendarType:** The calendar type for the site (2). This value MUST NOT be NULL.



**@WebAdjustHijriDays:** The number of days to extend or reduce the current month in Hijri calendars on the site (2) with which this alert subscription is associated. This value MUST NOT be NULL.

**@AlertTitle:** The title of the alert (1) of the alert subscription.

**@AlertType:** An **integer** whose lower eight bits specify the type of the alert (1), as specified in section 2.2.1.3. This MUST also contain the special alert (1) flags (section 2.2.2.2) for an always notify alert or a system alert. This value MUST NOT be NULL.

**@AlertTemplateName:** The name of the alert template of the alert subscription.

**@Filter:** The syntax query in CAML for a filter to apply to the alert (1) associated with this alert subscription.

**@BinaryFilter:** The compiled binary syntax query in CAML for a filter to apply to the alert (1) associated with this alert subscription.

**@Properties:** An XML blob representing the properties of the alert (1).

**@DeliveryChannel:** Specifies the delivery channel for the alert subscription as defined by the alert (1) delivery channel type (section 2.2.1.1).

**@UserEmail:** The e-mail address of the user.

**@ItemName:** The name of the item associated with the alert subscription.

**@ItemId:** The item identifier associated with the alert subscription. This MUST NOT be NULL if the **AlertType** of the alert subscription is an Item type (section 2.2.1.3) and MUST be NULL if the **AlertType** of the alert subscription is a List type.

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

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

Value	Description
"0"	Successful completion.
"31"	Error finding or modifying the alert subscription.

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

### 3.1.5.43 proc\_MoveDataToOccurrence

The **proc\_MoveDataToOccurrence** stored procedure is called to move meeting instance data to the specified occurrence. This may be used to fix orphaned instances. An occurrence can be orphaned when a series is instantiated and then the recurrence pattern is changed, making the **RecurrenceID** invalid. The user may use this stored procedure to preserve the data in the instance by transferring it to a valid occurrence. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_MoveDataToOccurrence (
    @UserId                int,
    @UserTitle             nvarchar(255),
    @SiteId                uniqueidentifier,
    @WebId                 uniqueidentifier,
    @FromInstanceID       int,
```

```

    @ToInstanceID          int,
    @CheckFromInstanceID  bit,
    @OverwriteToInstance  bit,
    @CreateMeetingEntry   bit,
    @ToDTStartUTC         datetime,
    @CreateOrphaned       bit,
    @RequestGuid           uniqueidentifier = NULL OUTPUT
);

```

**@UserId:** An identifier for the user that is requesting the operation. This value MUST refer to an existing user identifier for the specified site (2).

**@UserTitle:** A **string** containing the title of the user who wishes to perform the operation. This parameter is optional.

**@SiteId:** The site collection identifier of the site collection containing the meeting series.

**@WebId:** The site identifier of the site (2) containing the meeting series.

**@FromInstanceID:** The identifier of the instance to be moved.

**@ToInstanceID:** The identifier of the instance to which to be moved. If this parameter is "-3", the instance data is not copied to any target occurrence, but is removed from the source occurrence.

**@CheckFromInstanceID:** A bit parameter that specifies whether validation on **FromInstance** is performed or not. If this parameter is "1" and *@FromInstanceID* is the same as *@ToInstanceID*, it MUST return "2". If this parameter is "1" and *@FromInstanceID* is not the same as *@ToInstanceID*, validation on **FromInstance** is performed. If this parameter is zero ("0"), no validation is performed on **FromInstance**.

**@OverwriteToInstance:** A bit parameter that specifies whether the instance should be overwritten.

**@CreateMeetingEntry:** A bit parameter that specifies whether a new meeting entry should be created in the table. The value MUST be the opposite of *@OverwriteToInstance*.

**@ToDTStartUTC:** Start date of the meeting series of the target instance in UTC.

**@CreateOrphaned:** A bit parameter that specifies whether the target instance should be orphaned from the new series.

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

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

Value	Description
"0"	Successful completion.
"2"	The Source Instance was not found, was not orphaned, or contained invalid event (2) type data.
"3"	The list item corresponding to the From Instance does not exist.
"5"	Access denied. The <b>current user</b> specified by the <i>@UserId</i> parameter is not the same as the author of the list item of <b>FromInstance</b> when the instance restriction level is "1".
"13"	Data was not moved to the target occurrence because at least one of the following conditions

Value	Description
	<p>was met:</p> <ol style="list-style-type: none"> <li>1. Source and target instances were the same and <b>CheckFromInstanceID</b>, <b>OverwriteToInstance</b>, or <b>CreateMeetingEntry</b> is "1".</li> <li>2. <b>OverwriteToInstance</b> and <b>CreateMeetingEntry</b> are both "1" or <b>OverwriteToInstance</b> is "1" when <b>ToInstanceId</b> is "-3".</li> <li>3. The target instance was not found.</li> </ol>
"33"	Attempt to delete directories that contain <b>checked out</b> files in the <b>FromInstance</b> . It is returned by <b>proc_DropListRecord</b> . See <a href="#">[MS-WSSDLIM]</a> .
"87"	Invalid parameter encountered when creating the meeting series item. <i>@SiteId</i> or <i>@WebId</i> contain invalid data. Returned by <b>proc_InstantiateMtgSeriesOccurrence</b> (section <a href="#">3.1.5.39</a> ).
"1150"	Concurrency violation during <b>proc_DropListRecord</b> . See <a href="#">[MS-WSSDLIM]</a> .

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

### 3.1.5.44 **proc\_NavStructAddNewNodeByDocId**

The **proc\_NavStructAddNewNodeByDocId** stored procedure is called to add a new navigation node that points to an existing document in the site (2) from the site's navigation structure. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_NavStructAddNewNodeByDocId(
    @SiteId          uniqueidentifier,
    @WebId           uniqueidentifier,
    @Eid             int,
    @EidParent       int,
    @EidPrevSib     int,
    @ElementType    tinyint,
    @DocId           uniqueidentifier,
    @Name            nvarchar(256),
    @NameResource    nvarchar(520),
    @NodeMetaInfo    varbinary(max),
    @NonNavPage      bit,
    @NavSequence     bit,
    @EidBase         int,
    @@EidHome        int OUTPUT,
    @RequestGuid     uniqueidentifier = NULL OUTPUT
);

```

**@SiteId:** Specifies the site collection identifier of the site collection that contains the site (2).

**@WebId:** Specifies the site identifier of the site (2) to which the navigation node is to be added.

**@Eid:** Specifies the navigation node element identifier of the new navigation node. This MUST NOT be NULL. Values between "1" and "999" are treated as temporary values. The final navigation node element identifier of a temporary value MUST be the sum of *@Eid* and *@EidBase*. An *@Eid* value of "1000" MUST NOT be used for nodes other than the home page. The combination of the navigation node element identifier, *@SiteId*, and *@WebId* MUST be unique.

**@EidParent:** Specifies the navigation node element identifier of the navigation node under which the new navigation node is being added. This MUST NOT be NULL. Values between "1" and "999" are treated as temporary values. The final navigation node element identifier of a temporary value MUST be the sum of *@EidParent* and *@EidBase*.

**@EidPrevSib:** Specifies the navigation node that precedes the new node. This MUST be either the navigation node element identifier of the preceding node or a value from the following table:

Value	Description
"-1"	The new navigation node MUST be added as the first child.
"-2"	The new navigation node MUST be added as the last child.

Values between "1" and "999" are treated as temporary values. If *@EidPrevSib* specifies a navigation node element identifier, the final navigation node element identifier of a temporary value MUST be the sum of *@EidPrevSib* and *@EidBase*.

**@ElementType:** Specifies the navigation node type (section [2.2.1.5](#)) of the new navigation node. This value MUST be zero. This MUST NOT be NULL.

**@DocId:** The identifier of the document to which the new navigation node points.

**@Name:** The display name of the navigation node. This MUST NOT be NULL

**@NameResource:** The resource name for the navigation node.

**@NodeMetaInfo:** A binary serialization of the metadata for a navigation node in metadict form ([\[MS-FPSE\]](#) section 2.2.2.2.11). This MAY be NULL.

**@NonNavPage:** A bit specifying whether the navigation node is hidden when rendering the navigation structure. If this is "1", the navigation node SHOULD be omitted<sup><7></sup>. This MUST NOT be NULL.

**@NavSequence:** A bit specifying whether the node represents a link bar. This value MUST be zero. This MUST NOT be NULL.

**@EidBase:** The base value to use if the *@Eid* value is a temporary value. This MUST be the value returned by **proc\_NavStructAllocateEidBlockWebId** (section [3.1.5.46](#)) before this stored procedure is called. If the *@Eid* value is NOT a temporary value, this MUST be ignored.

**@@EidHome:** If the new navigation node points to the home page of the site (2) and the navigation node represented by *@EidParent* does not point to a link bar, this MUST return the navigation node element identifier of the new navigation node. Otherwise, it MUST return NULL.

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

**Return Values:** Returns an **integer** return code that MUST be listed in the following table:

Value	Description
"0"	Successful completion.
"10001"	Either <i>@SiteId</i> or <i>@WebId</i> is NULL.
"10002"	The navigation node specified by <i>@EidParent</i> does not exist.

Value	Description
"10006"	The new navigation node was not added.
"10007"	The document specified by <i>@DocId</i> does not exist.
"10008"	The document specified by <i>@DocId</i> has no published version.

**Result Sets:** MUST return either zero or two result sets. If the return code is zero ("0"), two result sets MUST be returned; otherwise no result set MUST be returned.

### 3.1.5.44.1 Nav Data Result Set

The Nav Data result set returns information about the newly added navigation node. If returned, this result set MUST return one row (3), as defined section [2.2.4.3](#).

### 3.1.5.44.2 Nav Children Result Set

Returns the navigation node element identifier information of all the child objects (1) of the new navigation node. If returned, this result set MUST contain no rows (3) as defined in section [2.2.4.4](#).

### 3.1.5.45 proc\_NavStructAddNewNodeByUrl

The **proc\_NavStructAddNewNodeByUrl** stored procedure is called to add a new navigation node that points to a specified URL to the site's (2) navigation structure (section [3.1.1.5](#)). The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_NavStructAddNewNodeByUrl (
    @SiteId                uniqueidentifier,
    @WebId                 uniqueidentifier,
    @Eid                   int,
    @EidParent             int,
    @EidPrevSib           int,
    @ElementType           tinyint,
    @Url                   nvarchar(260),
    @Name                  nvarchar(256),
    @NameResource          nvarchar(520),
    @NodeMetaInfo         varbinary(max),
    @IgnoreIfExists       bit,
    @NonNavPage            bit,
    @NavSequence           bit,
    @DateLastModified     datetime,
    @EidBase               int,
    @@EidHome              int OUTPUT
    @RequestGuid           uniqueidentifier = NULL OUTPUT
);

```

**@SiteId:** The site collection identifier of the site collection that contains the site (2).

**@WebId:** The site identifier of the site (2) to which the navigation node is to be added.

**@Eid:** Specifies the navigation node element identifier of the new navigation node. This MUST NOT be NULL. Values between "1" and "999" are treated as temporary values. The final navigation node element identifier of a temporary value MUST be the sum of *@Eid* and *@EidBase*. An *@Eid* value of "1000" MUST NOT be used for nodes other than the home page.

**@EidParent:** The navigation node element identifier of the navigation node under which the new navigation node is being added.

**@EidPrevSib:** The positioning of the new navigation node. This MUST either be the navigation node element identifier that precedes the new navigation node, or MUST be a value from the following table:

Value	Description
"-1"	The new navigation node MUST be added as the first child.
"-2"	The new navigation node MUST be added as the last child.

**@ElementType:** Specifies the navigation node type (section [2.2.1.5](#)) of the navigation node. This MUST NOT be NULL.

**@Url:** The URL of the navigation node being added. If *@ElementType* is zero ("0") and *@NavSequence* is zero ("0"), the navigation node is based on the identifier of a document in the site collection. If the navigation node is based on the **document identifier**, the URL MUST be in store-relative form.

**@Name:** The display name of the navigation node. This MUST NOT be NULL.

**@NameResource:** The resource name for the navigation node.

**@NodeMetainfo:** A binary serialization of the metadata for a navigation node in metadict form ([\[MS-FPSE\]](#) section 2.2.2.2.11). This MAY be NULL.

**@IgnoreIfExists:** A bit specifying whether to ignore adding the new navigation node if one specified by *@Eid* already exists. If "1" and a node with the same navigation node element identifier exists, the new navigation node MUST NOT be added. If set to zero ("0") and a navigation node with the same navigation node element identifier exists, the new navigation node is created as normal, but the *RankChild* value is incremented.

**@NonNavPage:** A bit specifying whether the navigation node is hidden when rendering the navigation structure. If this is "1", the navigation node SHOULD be omitted. This MUST NOT be NULL.

**@NavSequence:** A bit specifying whether the new navigation node represents a link bar. If this is "1", the navigation node MUST represent a link bar. This MUST NOT be NULL.

**@DateLastModified:** A datetime value specifying the time in UTC when this navigation node was added.

**@EidBase:** The base value to use if the *@Eid* value is a temporary value. This MUST be the value returned by **proc\_NavStructAllocateEidBlockWebId** (section [3.1.5.46](#)) before this stored procedure is called. If the *@Eid* value is not a temporary value, this MUST be ignored.

**@@EidHome:** If the new navigation node points to the home page of the site (2) and the navigation node represented by *@EidParent* does not point to a link bar, this MUST return the navigation node element identifier of the new navigation node. Otherwise, it MUST return NULL.

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

**Return Values:** Returns an **integer** return code that MUST be listed in the following table:

Value	Description
"0"	Successful completion.
"10001"	Either @SiteId or @WebId is NULL.
"10002"	The navigation node specified by @EidParent does not exist.
"10006"	The new navigation node was not added.
"10007"	The document specified by @Url does not exist.
"10008"	The document specified by @Url does not have a published version.

**Result Sets:** If the return code is zero ("0"), two result sets MUST be returned; otherwise no result set MUST be returned.

### 3.1.5.45.1 Nav Data Result Set

Returns information about the new added navigation node. If returned, this result set MUST return one row (3), as defined in section [2.2.4.3](#).

### 3.1.5.45.2 Nav Children Result Set

The **Nav Children** result set returns the navigation node element identifier information of all the child objects (1) of the new navigation node. If returned, this result set MUST contain no rows (3), as defined in section [2.2.4.4](#).

### 3.1.5.46 proc\_NavStructAllocateEidBlockWebId

The **proc\_NavStructAllocateEidBlockWebId** stored procedure is invoked to reserve a set of identifiers for navigation node elements for new navigation nodes or navigation nodes that are moved under a different navigation node. This stored procedure MUST update the site's (2) next available identifier for navigation node elements. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_NavStructAllocateEidBlockWebId(
    @WebId          uniqueidentifier,
    @TempEidCount   int,
    @EidMaxNew      int,
    @@EidBase       int OUTPUT,
    @RequestGuid    uniqueidentifier = NULL OUTPUT
);

```

**@WebId:** The identifier of the site (2) for which identifiers for navigation node elements are being reserved.

**@TempEidCount:** The number of new identifiers for navigation node elements that need to be reserved. This value MUST be greater than or equal to zero.

**@EidMaxNew:** Specifies the maximum value of a temporary identifier for a navigation node element for the navigation nodes that are added or moved in subsequent calls.

**@@EidBase:** An output parameter containing the base identifier for navigation node elements to be used by the subsequent calls to update the navigation structure (section [3.1.1.5](#)).

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

**Return Values:** Returns an **integer** return code that MUST be listed in the following table:

Value	Description
"0"	Successful completion.
"10001"	The site (2) specified by @WebId does not exist. @@EidBase is set to NULL.
"10004"	The next available identifier for navigation node elements value for this site (2) could not be updated.

**Result Sets:** MUST return zero result sets.

### 3.1.5.47 proc\_NavStructClear

The **proc\_NavStructClear** stored procedure is called to delete the navigation structure of a site and any references to the contained navigation nodes. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_NavStructClear (
    @SiteId          uniqueidentifier,
    @WebId           uniqueidentifier,
    @RequestGuid     uniqueidentifier = NULL OUTPUT
);
```

**@SiteId:** The site collection identifier of the site collection that contains the site (2).

**@WebId:** The site identifier of the site (2) for which the navigation structure is to be deleted.

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

**Return Values:** **proc\_NavStructClear** returns an **integer** return code that MUST be ignored.

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

### 3.1.5.48 proc\_NavStructDeleteNodeByIds

The **proc\_NavStructDeleteNodeByIds** stored procedure is called to delete a navigation node and all its child navigation nodes. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_NavStructDeleteNodeByIds (
    @SiteId          uniqueidentifier,
    @WebId           uniqueidentifier,
    @Eid            int,
    @DateModified    datetime,
    @@DateModifiedReally datetime OUTPUT,
    @RequestGuid     uniqueidentifier = NULL OUTPUT
);
```

**@SiteId:** The site collection identifier for the site collection that contains the site (2) specified by @WebId.

**@WebId:** The site identifier of the site (2) that contains the navigation node.



**@Eid:** The navigation node element identifier of the navigation node to be deleted.

**@DateModified:** A datetime value specifying the last known time in UTC when this navigation node was changed. The value of *@DateModified* MUST match the actual last-modified time to complete successfully.

**@@DateModifiedReally:** An output parameter containing a datetime value specifying the time in UTC when the navigation node was last changed. For a successful completion, this specifies the time when the navigation node was deleted. If the return value is "10004", this specifies the time when the navigation node was last changed. If the navigation node does not exist, this value SHOULD be NULL.

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

**Return Values:** Returns an **integer** return code that MUST be listed in the following table:

Value	Description
"0"	Successful completion or the navigation node does not exist.
"10004"	The navigation node has been changed since the time specified by <i>@DateModified</i> .

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

### 3.1.5.49 **proc\_NavStructLogChangesAndUpdateSiteChangedTime**

The **proc\_NavStructLogChangesAndUpdateSiteChangedTime** stored procedure is called to update the change log with information about site (2) navigation changes that took place at a specified time. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_NavStructLogChangesAndUpdateSiteChangedTime (  
    @SiteId                uniqueidentifier,  
    @WebId                 uniqueidentifier,  
    @TimeLastModifiedIncoming    datetime,  
    @RequestGuid           uniqueidentifier = NULL OUTPUT  
);
```

**@SiteId:** The site collection identifier of the site collection containing the change log.

**@WebId:** The site identifier of the site (2) containing the change log.

**@TimeLastModifiedIncoming:** The datetime of the last update. If this parameter is NULL, then the current datetime MUST be used in the change log.

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

**Return Values:** Returns an **integer** value that MUST be ignored.

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

### 3.1.5.50 **proc\_NavStructMoveNode**

The **proc\_NavStructMoveNode** stored procedure is called to move a navigation node and all of its child objects (1) from under one parent navigation node to another within the site (2). The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_NavStructMoveNode (
    @SiteId                uniqueidentifier,
    @WebId                  uniqueidentifier,
    @Eid                    int,
    @EidParentNew          int,
    @EidPrevSib            int,
    @DateModified          datetime,
    @EidBase                int,
    @EidHome                int,
    @@DateModifiedReally    datetime OUTPUT,
    @RequestGuid            uniqueidentifier = NULL OUTPUT
);

```

**@SiteId:** The site collection identifier of the site collection that contains the site (2) specified by *@WebId*.

**@WebId:** The site identifier of the site (2) that contains the navigation node.

**@Eid:** The navigation node element identifier of the navigation node to move.

**@EidParentNew:** The navigation node element identifier of the navigation node within the site (2) specified by *@SiteId* and *@WebId* that is the new parent of the navigation node being moved.

**@EidPrevSib:** Specifies the navigation node that precedes the new node. MUST be either the navigation node element identifier of the preceding node or a value from the following table:

Value	Description
-1	The new navigation node MUST be added as the first child.
-2	The new navigation node MUST be added as the last child.

**@DateModified:** A datetime value specifying the last known time when this navigation node was changed that must match the actual last-modified time for the operation to complete successfully.

**@EidBase:** The base value to use if the *@Eid* value is a temporary value. This MUST be the value returned by **proc\_NavStructAllocateEidBlockWebId** (section [3.1.5.46](#)) before this stored procedure is called. If the *@Eid* value is NOT a temporary value, this MUST be ignored.

**@EidHome:** The navigation node element identifier of the navigation node that points to the home page of the site (2) specified by *@SiteId* and *@WebId*.

**@@DateModifiedReally:** An output parameter containing a datetime value specifying the time when the navigation node was last changed. For a successful completion, this represents the time when the navigation node was moved. If the return code is "10004", this represents the time when the navigation node was last changed. If the navigation node does not exist or cannot be moved, this value SHOULD be NULL.

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

**Return Values:** Returns an **integer** return code that MUST be listed in the following table:

Value	Description
"0"	Successful completion.

Value	Description
"10004"	The navigation node specified by @Eid for the site (2) specified by @SiteId and @WebId has been changed since the time specified by @DateModified.
"10005"	The navigation node specified by @Eid does not exist for the site (2) specified by @WebId and @SiteId.
"10002"	The navigation node specified by @EidParentNew does not exist in the site (2) specified by @WebId and @SiteId
"10006"	The navigation node could not be moved.

**Result Sets:** If the return code is zero ("0"), this stored procedure MUST return a **Nav Data** result set and a **Nav Children** result set. Otherwise this procedure MUST NOT return any result sets.

### 3.1.5.50.1 Nav Data Result Set

Returns information about the new added navigation node. If returned, this result set MUST return one row (3), as defined in section [2.2.4.3](#).

### 3.1.5.50.2 Nav Children Result Set

The **Nav Children** result set returns the navigation node element identifier information of all the child objects (1) of the new navigation node. If returned, this result set MUST contain no rows (3), as defined in section [2.2.4.4](#).

### 3.1.5.51 proc\_OrphanRecurringEventExceptions

The **proc\_OrphanRecurringEventExceptions** stored procedure is called to orphan the meeting instances in the database that correspond to exceptions.

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

```

PROCEDURE proc_OrphanRecurringEventExceptions (
    @SiteID                uniqueidentifier,
    @ListID                uniqueidentifier,
    @WebID                 uniqueidentifier,
    @UID                   uniqueidentifier,
    @ServerTemplate        int,
    @SeriesItemID          int,
    @UserID                 int,
    @RecurrenceColName     nvarchar(64),
    @RecurrenceRowOrd      int = 0,
    @EventTypeColName      nvarchar(64),
    @EventTypeRowOrd       int = 0,
    @RecDataColName        nvarchar(64),
    @RecDataRowOrd         int = 0,
    @UIDColName            nvarchar(64),
    @UIDRowOrd             tinyint = 0,
    @RequestGuid           uniqueidentifier = NULL OUTPUT
);

```

**@SiteId:** The site collection identifier for the site collection containing the list (1) that the specified item is being added to.

**@ListID:** The list identifier of the list (1) that the specified item is being added to.

**@WebId:** The site identifier for the site (2) containing the list (1) that the specified item is being added to.

**@UID:** A GUID identifying a particular meeting series.

**@ServerTemplate:** The **integer** value of the meetings list template type.

**@SeriesItemID:** The identifier of the meeting instance list item corresponding to the master instance of the recurring meeting series.

**@UserID:** The **integer** that identifies the current user.

**@RecurrenceColName:** The name of the column (1) in the **UserData** view that corresponds to the recurrence identifier of the recurring meeting series.

**@RecurrenceRowOrd:** The ordinal of the list item that SHOULD contain the recurrence identifier specified inside the column (1) identified by the parameter *@RecurrenceColName*.

**@EventTypeColName:** The name of the column (1) in the **UserData** view that corresponds to the event (2) type. For more details about **UserData** view, see [\[MS-WSSFO2\]](#) section 2.2.7.8.

**@EventTypeRowOrd:** The ordinal of the list item that SHOULD contain the event (2) type information specified inside the column (1) identified by the parameter *@EventTypeColName*.

**@RecDataColName:** The name of the column (1) in the **UserData** view that corresponds to the recurrence data of the recurring meeting series.

**@ RecDataRowOrd:** The ordinal of the list item that SHOULD contain the recurrence data specified inside the column (1) identified by the parameter *@RecDataColName*.

**@UIDColName:** The name of the column (1) in the **UserData** view that corresponds to the unique identifier of the recurring meeting series.

**@UIDRowOrd:** The name of the row (3) ordinal in the **UserData** view that contains the unique identifier of the recurring meeting series in the column (1) specified by the *@UIDColName* parameter.

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

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

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

### 3.1.5.52 **proc\_ProcessDelMtgAttendeeListItem**

The **proc\_ProcessDelMtgAttendeeListItem** stored procedure is called to remove a meeting attendee list item from a meetings list (1). The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ProcessDelMtgAttendeeListItem(
    @SiteId          uniqueidentifier,
    @WebId           uniqueidentifier,
    @ListId          uniqueidentifier,
    @ItemId          int,
    @RequestGuid     uniqueidentifier = NULL OUTPUT
);
```

**@SiteId:** The site collection identifier of a site collection.

**@WebId:** The site identifier of a site (2) within the site collection.

**@ListId:** The list identifier of an attendees list (1) in the site (2).

**@ItemId:** The identifier of the list item corresponding to the item for an attendee.

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

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

Value	Description
"0"	Successful completion.
"13"	Attendee was not found in list (1).

**Result Sets:** **MUST** return zero or one result sets as defined in the following subsection.

### 3.1.5.52.1 Site Audit Mask Result Set

The **Site Audit Mask** result set returns the information about the **audit flags** associated with the specified site (2). The **Site Audit Mask** result set is returned if and only if a single attendee match is found, and there exists a site (2) defining the security roles on the specified site (2) designated by the **@WebId** parameter. The **Site Audit Mask** result set **MUST** return a single row (3).

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

```
{Id}                uniqueidentifier,  
{AuditFlags}       int,  
{InheritAuditFlags} int,  
{GlobalAuditMask} int;
```

**{Id}:** The identifier of the specified site (2). This **MUST** be the same as **@WebId**.

**{AuditFlags}:** An audit flag's value determining the operations to be tracked on this site (2). This value **MUST** be NULL if the site (2) cannot be found within the site collection.

**{InheritAuditFlags}:** An audit flag's value determining the operations to be tracked on one of this site's (2) child objects (1). This value **MUST** be NULL if the site (2) cannot be found within the site collection.

**{GlobalAuditMask}:** An audit flag's value determining the operations to be tracked across the site collection that contains this site (2). This value **MUST** be NULL if the site collection cannot be found.

### 3.1.5.53 proc\_PutWebNavStructNode

The **proc\_PutWebNavStructNode** stored procedure is called to update the properties of a navigation node. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_PutWebNavStructNode (  
    @SiteId                uniqueidentifier,  
    @WebId                 uniqueidentifier,  
    @Eid                   int,  
    @EidBase               int,
```

```

    @EidHome                int,
    @DateModified           datetime = NULL,
    @DirName                nvarchar(256) = NULL,
    @LeafName               nvarchar(128) = NULL,
    @UrlExternal            nvarchar(260) = NULL,
    @Name                   nvarchar(256) = NULL,
    @SetMetaInfo            bit = 0,
    @NodeMetainfo           varbinary(max) = NULL,
    @NonNavPage             bit = NULL,
    @NavSequence            bit = NULL,
    @WantOldName            bit = 0,
    @@DateModifiedReally    datetime = OUTPUT,
    @RequestGuid            uniqueidentifier = NULL OUTPUT
);

```

**@SiteId:** The site collection identifier of the site collection that contains the site (2) specified by *@WebId*.

**@WebId:** The site identifier of the site (2) to which the navigation node belongs.

**@Eid:** The navigation node element identifier of the navigation node to be updated.

**@EidBase:** The base value to use if the *@Eid* value is a temporary value. This MUST be the value returned by **proc\_NavStructAllocateEidBlockWebId** (section [3.1.5.46](#)) before this stored procedure is called. If the *@Eid* value is NOT a temporary value, this MUST be ignored.

**@EidHome:** The navigation node element identifier of the navigation node that points to the home page of the site (2).

**@DateModified:** A datetime value specifying the last known time when this navigation node was changed. If not NULL, the value of *@DateModified* MUST match the actual last-modified time to complete successfully. If NULL, the navigation node is considered current.

**@DirName:** The store-relative form directory name of the document that this navigation node represents. This MUST be ignored if *@LeafName* is NULL or the navigation node has a navigation node type "1" (section [2.2.1.5](#)). If NULL, the navigation node **Url** and **DocId** MUST NOT be changed.

**@LeafName:** The store-relative form leaf name of the document this navigation node represents. This MUST be ignored if *@DirName* is NULL or the navigation node has a navigation node type "1" (section [2.2.1.5](#)). If NULL, the navigation node **Url** and **DocId** MUST NOT be changed.

**@UrlExternal:** The URL that this navigation node represents. This MUST be ignored if the navigation node has a navigation node element **type** equal to zero (section [2.2.1.5](#)). If NULL, the navigation node **Url** MUST NOT be changed.

**@Name:** The display name of the navigation node. If this is NULL, then the display name MUST NOT be changed.

**@SetMetaInfo:** A bit specifying whether to update the metadata for a navigation node. If this is "1", the metadata MUST be set to the value specified by *@NodeMetainfo*. Otherwise, the metadata MUST NOT be changed.

**@NodeMetainfo:** A binary serialization of the metadata for a navigation node in metadict form ([\[MS-FPSE\]](#) section 2.2.2.2.11). This MAY be NULL.

**@NonNavPage:** A bit specifying whether the navigation node will be hidden when rendering the navigation structure. If this is "1", the navigation node SHOULD be omitted. If this is NULL, then the corresponding setting for the navigation node MUST NOT be changed.

**@NavSequence:** A bit specifying whether the node represents a link bar. If this navigation node represents a link bar, this bit MUST be "1". Otherwise, it MUST be zero. If the navigation node specified by *@Eid* has a navigation node element type equal to zero, this value MUST NOT be "1".

**@WantOldName:** A bit specifying whether the caller needs the old display name of the navigation node. If this is "1", the **Old Name** result set MUST be returned.

**@@DateModifiedReally:** An output parameter containing a datetime value that specifies the time when the navigation node was last changed. For a successful completion, this represents the time when the navigation node was updated. If the return code is "10004", this represents the time when the navigation node was last changed. If the navigation node does not exist or cannot be updated, this value SHOULD be NULL.

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

**Return Values:** Returns an **integer** return code that MUST be listed in the following table:

Value	Description
"0"	Successful completion.
"10005"	The navigation node specified by <i>@Eid</i> does not exist for the site (2) specified by <i>@WebId</i> and <i>@SiteId</i> .
"10004"	The navigation node has been changed since the time specified by <i>@DateModified</i> .
"10007"	The document specified by <i>@DirName</i> and <i>@LeafName</i> does not exist.
"10006"	The navigation node was not updated.

**Result Sets:** MUST return zero, two, or three result sets. If the return code is anything other than zero ("0"), no result set MUST be returned. If the return code is zero ("0"), either two or three result sets MUST be returned.

### 3.1.5.53.1 Old Name Result Set

The **Old Name** result set returns the old name of the navigation node. If the *@WantOldName* bit is set to "1" and the return code is zero ("0"), this result set MUST be returned; otherwise this result set MUST NOT be returned. If the result set is returned, it MUST return one row (3). The T-SQL syntax for the result set is as follows:

```
{OldName}          nvarchar(256);
```

**{OldName}:** The display name of the navigation node before it was changed by this stored procedure.

### 3.1.5.53.2 Nav Data Result Set

The **Nav Data** result set returns information about the updated navigation node. If returned, this result set MUST return one row (3), as defined in section [2.2.4.3](#).

### 3.1.5.53.3 Nav Children Result Set

The **Nav Children** result set returns the navigation node element identifier information of all the child objects (1) of the new navigation node. If returned, this result set MUST contain zero or more rows (3), as defined in section [2.2.4.4](#).

### 3.1.5.54 proc\_RelinkMeeting

The **proc\_RelinkMeeting** stored procedure is called to relink a meeting instance with a meeting series. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_RelinkMeeting(  
    @MeetingsListID    uniqueidentifier,  
    @InstanceID        int,  
    @UserId             int,  
    @RequestGuid       uniqueidentifier = NULL OUTPUT  
);
```

**@MeetingsListID:** A GUID referring to the meetings list (1).

**@InstanceID:** The identifier of the instance of the meeting that will be re-linked.

**@UserId:** An **integer** value referring to the user performing the action.

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

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

Value	Description
"0"	Successful completion.
"2"	No meeting instances were updated.

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

### 3.1.5.55 proc\_ResolveWikiLinkFile

The **proc\_ResolveWikiLinkFile** stored procedure attempts to determine the full URL to a file from a server-relative URL. Three server-relative URLs are passed in, and the first one to succeed (if any) will return the full URL of the file. The stored procedure is defined using T-SQL syntax, as follows:

```
PROCEDURE proc_ResolveWikiLinkFile(  
    @SiteId            uniqueidentifier,  
    @LinkId            int,  
    @FileUrl           nvarchar(V_STORE_MAX_FULLURL),  
    @FolderUrl         nvarchar(V_STORE_MAX_FULLURL),  
    @FileUrl2          nvarchar(V_STORE_MAX_FULLURL),  
    @RequestGuid       uniqueidentifier = NULL OUTPUT  
);
```

**@SiteId:** The GUID of the site (2) containing the file specified by *@FileUrl*, *@FolderUrl*, *@FileUrl2*.

**@LinkId:** This parameter is passed back in the result set unmodified, and otherwise ignored.



**@FileUrl:** The first attempt of the server-relative URL, with no leading forward slash (/), to the file.

**@FolderUrl:** The second attempt of the server-relative URL, with no leading forward slash (/), to the file.

**@FileUrl2:** The third attempt of the server-relative URL, with no leading forward slash (/), to the file.

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

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

**Result Sets:** MUST return the result set defined in the following subsection.

### 3.1.5.55.1 Resolve Wiki Link File Result Set

The **Resolve Wiki Link** result set is defined using T-SQL syntax, as follows:

```
LinkId          int;
ResolvedUrl     nvarchar(V_STORE_MAX_FULLURL);
```

**LinkId:** This MUST be the same value as *@LinkId*.

**ResolvedUrl:** The full URL to the file indicated by *@FileUrl*, *@FolderUrl*, or *@FileUrl2*, if the file exists. Otherwise, it MUST be NULL.

### 3.1.5.56 proc\_ResolveWikiLinkItem

The **proc\_ResolveWikiLinkItem** stored procedure is called to determine the URL to a list item. The stored procedure is defined using T-SQL syntax, as follows:

```
PROCEDURE proc_ResolveWikiLinkItem(
    @LinkId          int,
    @ListId          uniqueidentifier,
    @ItemId          int,
    @RequestGuid     uniqueidentifier = NULL OUTPUT
);
```

**@LinkId:** This parameter is passed back in the result set unmodified, and otherwise ignored.

**@ListId:** The GUID of the list (1) containing the list item specified by *@ItemId*.

**@ItemId:** The identifier of the list item within the list specified by *@ListId*.

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

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

**Result Sets:** MUST return the result set defined in the following subsection.

### 3.1.5.56.1 Resolve Wiki Link Item Result Set

The **Wiki Link Item** result set is defined using T-SQL syntax, as follows:

```
LinkId          int;
```

```
ResolvedUrl nvarchar(V_STORE_MAX_FULLURL);
```

**LinkId:** This MUST be the same value as *@LinkId*.

**ResolvedUrl:** The full URL to the list item indicated by *@ItemId* in the list (1) indicated by *@ListId*, if the list item exists. Otherwise, it MUST be NULL.

### 3.1.5.57 **proc\_SetCommentAttribs**

The **proc\_SetCommentAttribs** stored procedure is called to update properties on a specified Web discussion comment. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_SetCommentAttribs(  
    @SiteId          uniqueidentifier,  
    @DocFullUrl     nvarchar(260),  
    @Id             int,  
    @Subject        nvarchar(255),  
    @Status         smallint,  
    @Comment        nvarchar(max),  
    @Size           int,  
    @UserTitle      nvarchar(255),  
    @RequestGuid    uniqueidentifier = NULL OUTPUT  
);
```

**@SiteId:** The site collection identifier for the site collection containing the specified document. *@SiteId* MUST NOT be NULL.

**@DocFullUrl:** The store-relative URL to the document.

**@Id:** The identifier of the Web discussion comment to be updated.

**@Subject:** The subject of this Web discussion comment. If NULL, the subject MUST NOT be changed.

**@Status:** A Web Discussion Comment Status Flag (section [2.2.2.3](#)) that indicates the status of a Web discussion comment. If NULL, the status MUST NOT be changed.

**@Comment:** The new body text of the Web discussion comment. If NULL, the body text MUST NOT be changed.

**@Size:** The new size in bytes of this Web discussion comment that is available for use in quota management. If NULL, the size MUST NOT be changed.

**@UserTitle:** The display name of the user that is adding the Web discussion comment.

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

**Return Values:** Returns an **integer** return code that MUST be in the following table:

Value	Description
"0"	Successful completion.
"3"	The specified Web discussion comment or document does not exist.

Value	Description
"212"	The specified site collection is locked, and writes are disallowed.
"1816"	The quota for the specified site collection was exceeded.

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

### 3.1.5.58 `proc_SetSecondaryResourceAtScope`

The **`proc_SetSecondaryResourceAtScope`** stored procedure is called to update a user resource string specified by the `@WebId`, `@ListId` and `@ResourceName` parameters. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_SetSecondaryResourceAtScope (
    @WebId          uniqueidentifier,
    @ListId         uniqueidentifier,
    @ResourceName   nvarchar(520),
    @BitType        bit,
    @LCID           int,
    @NVarCharVal    nvarchar(256),
    @NtextVal       nvarchar(max)
);

```

**@WebId:** The identifier of the site (2) that contains the list (1) indicated by `@ListID`. This MUST NOT be NULL.

**@ListId:** The identifier of the list (1) that contains the resource indicated by `@ResourceName`. This MUST NOT be NULL.

**@ResourceName:** The name of the resource to be updated. If the resource string with the name specified by this parameter already exists for the list (1) specified by `@WebId`, `@ListId` and `@LCID`, This procedure MUST NOT update the resource string. This MUST NOT be NULL.

**@BitType:** Type of the resource as defined in user resource type (section [2.2.1.10](#))

**@LCID:** The LCID for the user resource string. If `@LCID` is equal to the LCID of the site (2) specified by the `@WebId` parameter, then this stored procedure MUST NOT update the resource string.

**@NVarCharVal:** Specifies the value of the user resource string. If the value of `@BitType` is "1", this parameter MUST be NULL.

**@NtextVal:** Specifies the value of the user resource string. If the value of `@BitType` is zero ("0"), this parameter MUST be NULL.

**Return values:** MUST return an **integer** that MUST be zero.

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

### 3.1.5.59 `proc_SetVersionIndependentMetaInfo`

The **`proc_SetVersionIndependentMetaInfo`** stored procedure is invoked to set the version-independent metadata of a document. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_SetVersionIndependentMetaInfo (

```

```

@SiteId          uniqueidentifier,
@DirName         nvarchar(256),
@LeafName       nvarchar(128),
@MetaInfo       varbinary(max),
@MetaInfoSize   int,
@UpdateHCD      int,
@HCD            int,
@UpdatedVersion int OUTPUT,
@RequestGuid    uniqueidentifier = NULL OUTPUT
);

```

**@SiteId:** The site collection identifier of a site collection containing the document whose version-independent metadata is set.

**@DirName:** The directory name of the document whose version-independent metadata is set.

**@LeafName:** The leaf name of the document whose version-independent metadata is set.

**@MetaInfo:** A metadict holding all version-independent metadata for the document.

**@MetaInfoSize:** The size in bytes of the *@MetaInfo* parameter.

**@UpdateHCD:** An **integer** value specifying whether *@HCD, Has Copy Destination, parameter is ignored or not*. If this parameter is NULL or equal to 0, *@HCD* parameter MUST be ignored. Otherwise, *@HCD* parameter MUST not be ignored.

**@HCD:** If this parameter is zero ("0"), **tp\_HasCopyDestinations** ([\[MS-WSSDM2\]](#) section 3.1.4.30.1) of the corresponding list (1) that contains the document specified by the *@DirName* and *@LeafName* MUST be zero. Otherwise, it MUST be "1".

**@UpdatedVersion:** The new version of the metadata.

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

**Return Values:** Returns an **integer** return code that MUST be listed in the following table:

Value	Description
"0"	Successful completion.
"1816"	The site collection is over quota.
"212"	The site collection is locked.
"33"	The metadata has been concurrently updated while the stored procedure is executing.

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

### 3.1.5.60 **proc\_SetWebNavStructInheritance**

The **proc\_SetWebNavStructInheritance** stored procedure is invoked to update a site's (2) inheritable navigation structure to be inherited or unique. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_SetWebNavStructInheritance(
    @WebId          uniqueidentifier,

```

```

    @Inherits          bit,
    @@NavParentWebId  uniqueidentifier OUTPUT,
    @RequestGuid       uniqueidentifier = NULL OUTPUT
);

```

**@WebId:** The site identifier whose inheritable navigation structure is to be updated.

**@Inherits:** A bit value specifying whether the site (2) specified by *@WebId* inherits navigation structure from its **parent site** or not. When set to "1", the site (2) specified by *@WebId* MUST inherit its inheritable navigation structure from its parent site. When set to zero, the site (2) specified by *@WebId* MUST have a unique inheritable navigation structure.

**@@NavParentWebId:** Returns the GUID of the first site (2) in the site (2) hierarchy specified by *@WebId* that does not inherit navigation. When *@Inherits* is zero, MUST return NULL.

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

**Return Values:** Returns an **integer** return code that MUST be listed in the following table:

Value	Description
"0"	Successful completion.
"1"	The site (2) specified by @WebId does not exist, OR The site (2) specified by @WebId already has its inheritable navigation structure in the correct state as specified by the @Inherits flag, and no changes were made.

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

### 3.1.5.61 proc\_UnOrphanMtgOccurrence

The **proc\_UnOrphanMtgOccurrence** stored procedure is called to re-associate a meeting instance within a meeting series. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_UnOrphanMtgOccurrence (
    @WebId          uniqueidentifier,
    @InstanceID     int,
    @UserId         int,
    @RequestGuid    uniqueidentifier = NULL OUTPUT
);

```

**@WebId:** The site identifier of a Meeting Workspace site. The value MUST be a GUID.

**@InstanceID:** The identifier of the instance of the meeting that will be re-associated to the meeting series.

**@UserId:** The identifier of the user performing the action.

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

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

Value	Description
"0"	Successful completion.
"2"	Either a meeting instance with <i>@InstanceID</i> was not found, or <i>@InstanceID</i> is mapped to more than one row (3) in the <b>UserData</b> view. For more details about <b>UserData</b> view, see <a href="#">[MS-WSSFO2] section 2.2.7.8</a> .
"13"	No meetings list (1) was found on the Meeting Workspace site specified by <i>@WebId</i> .

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

### 3.1.5.62 **proc\_UpdateCachedNav**

The **proc\_UpdateCachedNav** stored procedure is called to update the navigation structure cache. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_UpdateCachedNav (
    @WebId                uniqueidentifier,
    @NavParentWebId       uniqueidentifier,
    @CachedNavData        varbinary(max),
    @CachedInheritedNavData varbinary(max),
    @CachedNavScopeData   nvarbinary(max),
    @NavParentCachedNavData varbinary(max),
    @NavParentCachedScopeData nvarbinary(max),
    @CachedNavDirty        int,
    @CachedDataVersion     int,
    @NavParentCachedDataVersion int,
    @RequestGuid           uniqueidentifier = NULL OUTPUT
);

```

**@WebId:** The site identifier for which the navigation structure cache is to be updated.

**@NavParentWebId:** The site identifier of the navigational parent site.

**@CachedNavData:** The node structure in binary data format from the navigation structure cache. The binary format is specified in section [2.2.5.1](#).

**@CachedInheritedNavData:** The node structure in binary data format from the navigational parent site.

**@CachedNavScopeData:** A string that specifies all the distinct security scope identifiers of all the navigation nodes of the site (2). Every element is included between single quotes and is separated by commas from the rest. The general format is specified as follows:

```
( '<Security Scope Identifier>', '<Security Scope Identifier>', <...> )
```

**@NavParentCachedNavData:** The node structure in binary data format of the cached navigation of the parent site.

**@NavParentCachedScopeData:** The string that contains all the distinct security scope identifiers of all the navigation nodes of the navigational parent site. The general format is specified as follows:

```
( '<Security Scope Identifier>', '<Security Scope Identifier>', <...> )
```

**@CachedNavDirty:** Specifies whether the navigation nodes contain outdated data. This MUST be 0.

**@CachedDataVersion:** The **integer** that defines the version of the cached data. This value MUST be equal to the current cached version for the navigational parent site specified by *@WebId* for an update to succeed. Each successful update increments the version by 1.

**@NavParentCachedDataVersion:** The **integer** that defines the version of the cached data for the navigational parent site. This value MUST be equal to the current cached version for the navigational parent site specified by *@NavParentWebId* for an update of the parent cache to succeed. Each successful update increments the version by 1.

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

**Return Values:** Returns an **integer** return code that MUST be in the following table:

Value	Description
"0"	Successful completion.
"1150"	Concurrency violation or an unknown error occurred. The cached data could not be updated.

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

### 3.1.5.63 `proc_UpdateItemJunctionsCurrentVersion`

The **`proc_UpdateItemJunctionsCurrentVersion`** stored procedure is called to update the **UIVersion** of a junction entry for a specific site collection, directory name, leaf name and level. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_UpdateItemJunctionsCurrentVersion(  
    @SiteId          uniqueidentifier,  
    @DirName         nvarchar(256),  
    @LeafName        nvarchar(128),  
    @NewUIVersion    int,  
    @Level           tinyint = 1,  
    @RequestGuid     uniqueidentifier = NULL OUTPUT  
);
```

**@SiteId:** Specifies the site collection identifier of the site collection in which the subscribed-to event (2) exists.

**@DirName:** Specifies the directory name of the subscribed event (2).

**@LeafName:** Specifies the leaf name, or file name, of the subscribed event (2).

**@NewUIVersion:** Specifies the **integer** to which the UI versions of junction entry specified by *@SiteId*, *@DirName*, *@LeafName* and *@Level* MUST be updated.

**@Level:** Specifies the status of the document for the subscribed event (2). The value MUST be listed in the following table:

Value	Description
"255"	Checked out
"2"	Draft

Value	Description
"1"	Publish
"0"	Unused

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

**Return Values:** Returns an **integer** that MUST be zero.

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

### 3.1.5.64 `proc_UpdateListNavNode`

The **proc\_UpdateListNavNode** stored procedure is called to update the navigation node that represents a list (1) in the navigation structure of a site (2). The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_UpdateListNavNode (
    @WebId                uniqueidentifier,
    @ListId                uniqueidentifier,
    @ViewId               uniqueidentifier,
    @Name                  nvarchar(256),
    @NavBarEid            int,
    @AddIfNotThere        bit,
    @UseRootFolderForNav  bit,
    @RequestGuid           uniqueidentifier = NULL OUTPUT
);

```

**@WebId:** The site identifier of the site (2) that contains the list (1) specified by *@ListId*.

**@ListId:** The list identifier of the list (1) whose navigation node is to be updated.

**@ViewId:** The identifier of the **default list view** of the list (1).

**@Name:** The display name of the navigation node.

**@NavBarEid:** The parent navigation node element identifier of the list (1) navigation node.

**@AddIfNotThere:** A bit specifying whether to add a new navigation node if no corresponding navigation node exists. This value MUST be zero.

**@UseRootFolderForNav:** A bit specifying whether to point the navigation node to the default list view of the list (1) or to the list's (1) root folder. This value MUST be zero.

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

**Return Values:** Returns an **integer** return code that MUST be zero.

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

### 3.1.5.65 `proc_UpdateSchedSubscriptionTimes`

The **proc\_UpdateSchedSubscriptionTimes** stored procedure is called to update the next notification times of Scheduled Subscriptions that have already passed a specific date. The T-SQL syntax for the stored procedure is as follows:



```

PROCEDURE proc_UpdateSchedSubscriptionTimes (
    @EventTime          datetime,
    @RequestGuid        uniqueidentifier = NULL OUTPUT
);

```

**@EventTime:** The UTC datetime of the event (2) relative to which the alert subscriptions are updated.

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

**Return Values:** MUST return an **integer** that MUST be zero.

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

### 3.1.5.66 proc\_UpdateUserResource

The **proc\_UpdateUserResource** stored procedure is called to update an existing user resource string or add a user resource string if the resource does not exist. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_UpdateUserResource (
    @WebId              uniqueidentifier,
    @ListId             uniqueidentifier,
    @ResourceName       nvarchar(520),
    @ResourceCategory   int,
    @BitType            bit,
    @BitDirty           bit,
    @LCID               int,
    @NVarCharVal        nvarchar(256),
    @NtextVal           nvarchar(max)
);

```

**@WebId:** The identifier of the site (2) that contains the user resource string. It MUST NOT be NULL.

**@ListId:** The identifier of the list (1) that contains the user resource string. If the resource string does not belong to a list (1), it MUST be an empty GUID. It MUST NOT be NULL.

**@ResourceName:** The name of the user resource string. It MUST NOT be NULL.

**@ResourceCategory:** The category of the user resource string. It MUST be one of the following values:

Value	Description	Constraint on the @ResourceName
"0"	The resource string is specified by the end user.	The value of @ResourceName MUST begin with digit or alphabetic letter.
"1"	The resource string is for the title of the site (2) specified by @WebId.	The value of @ResourceName MUST be "_WebTitle".
"2"	The resource string is for the description of the site (2) specified by @WebId.	The value of @ResourceName MUST be "_WebDescription".
3	The resource string is for	The value of @ResourceName MUST be "_ListTitle".

Value	Description	Constraint on the @ResourceName
	title of the list (1) specified by the @ListId.	
4	The resource string is for the description of the list (1) specified by the @ListId.	The value of @ResourceName MUST be "_ListDescription".
5	The resource string is for the title of a navigation node.	The value of @ResourceName MUST begin with string "_NavNode" and then be followed by the string representation of the navigation node element identifier. For example, if the navigation node element identifier is "1002", its value MUST be "_NavNode1002".

**@BitType:** Type of the resource as defined in user resource type (section [2.2.1.10](#))

**@BitDirty:** It MUST be NULL.

**@LCID:** The LCID for the user resource string specified by @NVarCharVal when @BitType is zero ("0") or @NtextVal when @BitType is "1". It MUST NOT be NULL.

**@NVarCharVal:** Value of the user resource string if @BitType is zero ("0"). It MUST be NULL if @BitType is "1".

**@NtextVal:** Value of the user resource string if @BitType is "1". It MUST be NULL if @BitType is zero ("0").

**Return values:** Returns an **integer** that MUST be zero.

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

### 3.1.6 Timer Events

If the connection timeout event (2) is triggered, the stored procedure is terminated and the call fails.

### 3.1.7 Other Local Events

No other local events (2) impact the operation of this protocol.

## 3.2 WSSEUX Client Details

The front-end Web server acts as a client when it calls the back-end database server requesting execution of stored procedures.

### 3.2.1 Abstract Data Model

### 3.2.2 Timers

A connection timeout timer is set up on the front-end Web server to govern the total connection time for any requests to the back-end database server. The amount of time is governed by a timeout value configured on the front-end Web server for all back-end database server connections.

### 3.2.3 Initialization

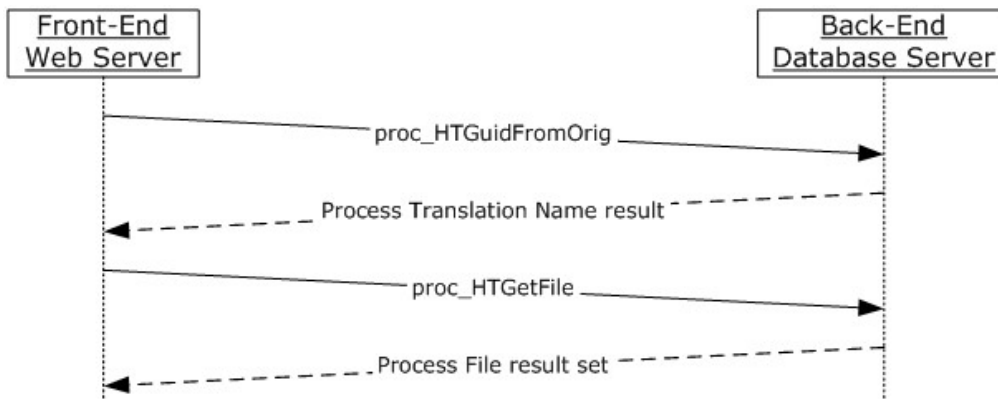
The front-end Web server MUST validate the user making the request before calling the stored procedures. The site collection identifier and the user identifier for the user making the request are looked up by the front-end Web server before calling additional stored procedures.

### 3.2.4 Higher-Layer Triggered Events

The front-end Web server handles each stored procedure with the same processing method of calling the stored procedure and waiting for the Result Code and any result sets that are returned.

The front-end Web server can execute dynamically generated SQL queries against the stored procedures, or the tables and views used within the database. However, unless otherwise specified, any data addition, removal, or modification MUST occur only by calling the listed stored procedures. SQL queries MUST NOT attempt to add, remove, or update data in any table or view in the Content or Configuration databases, unless explicitly described in this section.

#### 3.2.4.1 Fetch HTML Translate Cache Item

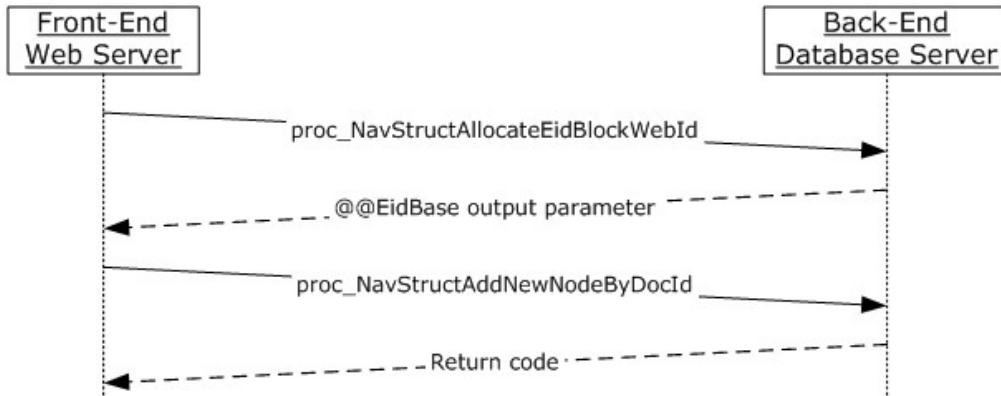


**Figure 2: Retrieve the HTML translate cache item**

This sequence gets invoked to fetch the item from the HTML translate cache for a document. Fetch HTML Translate Cache Item Sequence executes the following sequence of sprocs:

1. Call **proc\_HTGuidFromOrig**.
2. Process the **Translation Name** result set and get the translation name for the document.
3. Call **proc\_HTGetFile** with the translation name obtained from step 2.

### 3.2.4.2 CreateNavigationNodeByDocId

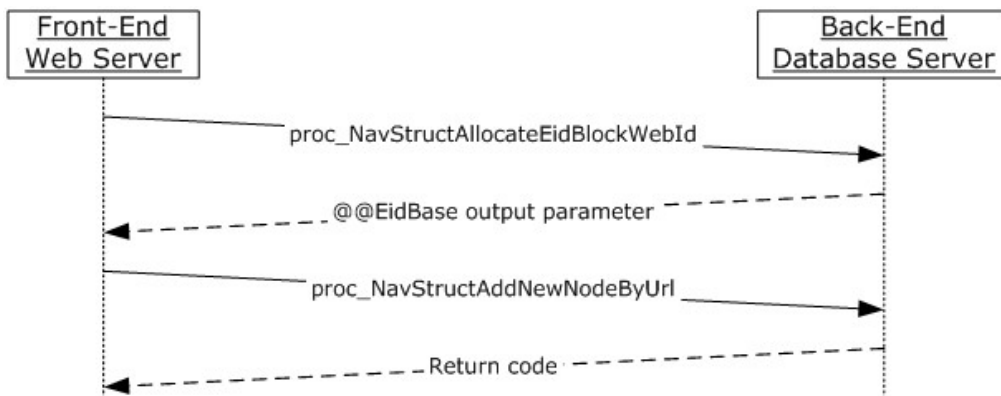


**Figure 3: CreateNavigationNodeByDocId operation**

**CreateNavigationNodeByDocId** is called to create a navigation node that points to an object (1) in the site (2). **CreateNavigationNodeByDocId** executes the following sequence of sprocs:

1. Call **proc\_NavStructAllocateEidBlockWebId**.
2. Process the output parameter **@@EidBase**.
3. Call **proc\_NavStructAddNewNodeByDocId**, passing in the identifier of the document and **@@EidBase** obtained in step 2.

### 3.2.4.3 CreateNavigationNodeByUrl



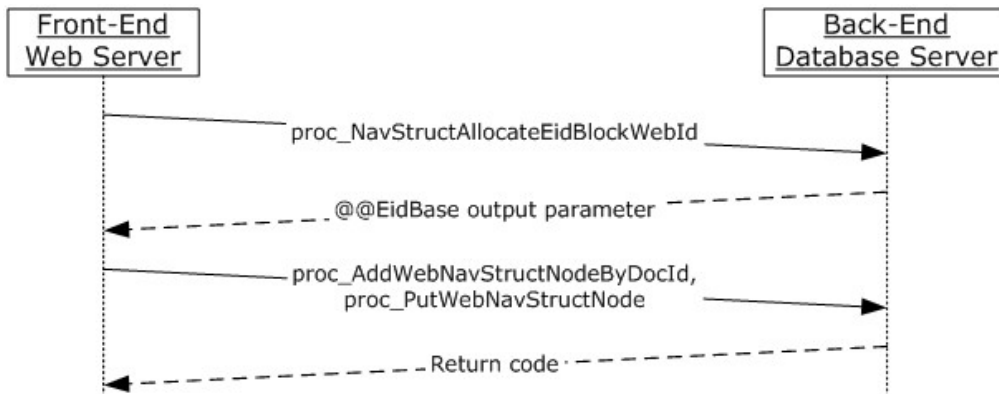
**Figure 4: CreateNavigationNodeByUrl operation**

**CreateNavigationNodeByUrl** gets invoked for creating a navigation node that points to a URL. **CreateNavigationNodeByUrl** executes the following sequence of sprocs:

1. Call **proc\_NavStructAllocateEidBlockWebId**.
2. Process the output parameter **@@EidBase**.

3. Call **proc\_NavStructAddNewNodeByUrl** passing in the URL and @@EidBase, which was obtained in step 2.

### 3.2.4.4 AddAndUpdateNavigationNode

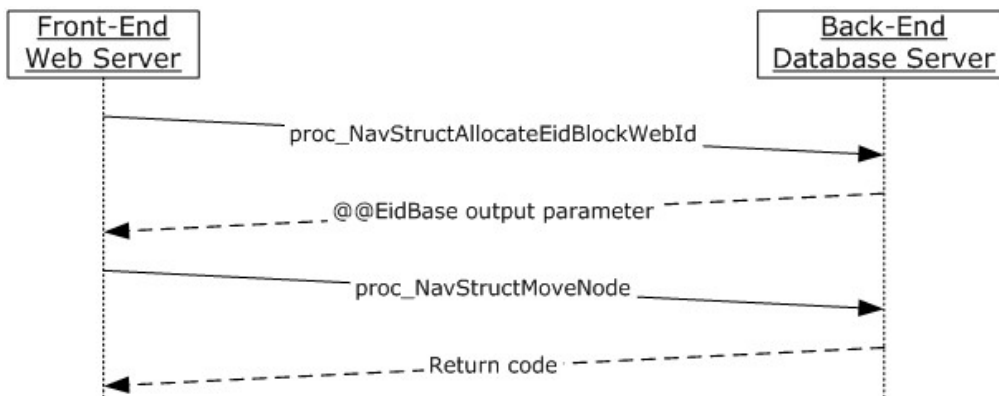


**Figure 5: AddAndUpdateNavigationNode operation**

**AddAndUpdateNavigationNode** gets invoked for creating and updating a navigation node in one transaction. **AddAndUpdateNavigationNode** executes the following sequence of sprocs:

1. Call **proc\_NavStructAllocateEidBlockWebId**.
2. Process the output parameter @@EidBase.
3. Call **proc\_AddWebNavStructNodeByUrl** or **proc\_AddWebNavStructNodeByDocID** and **proc\_PutWebNavStructNode** using @@EidBase, which was obtained in step 2.

### 3.2.4.5 MoveNavigationNode



**Figure 6: MoveNavigationNode operation**

**MoveNavigationNode** gets invoked for moving a navigation node. **MoveNavigationNode** executes the following sequence of sprocs:

1. Call **proc\_NavStructAllocateEidBlockWebId**.

2. Process the output parameter @@EidBase.
3. Call **proc\_NavStructMoveNode** passing in the identifier of the document and @@EidBase, which was obtained in step 2.

### **3.2.5 Message Processing Events and Sequencing Rules**

None.

### **3.2.6 Timer Events**

If the connection timeout event (2) is triggered, the connection and the stored procedure call fails.

### **3.2.7 Other Local Events**

No other local events (2) impact the operation of this protocol.

## 4 Protocol Examples

### 4.1 TreeView Operations

This example describes the requests made and responses returned between the front-end Web server and back-end database server when an end user expands tree nodes on the **TreeView** (section [3.1.1.6](#)) of a site (2).

In this scenario, the site (2) contains a subsite and a document library containing folders.

1. When the client requests the information for the site (2), the front-end Web server gets the information of the subsites and the lists (1) in the site (2) by calling **proc\_GetWebAndChildrenNSInfo** (section [3.1.5.27](#)) to the back-end database server.
2. The back-end database server returns one **SiteNSInfo** result set containing the subsites of the expanded site (2) node and one **ListNSInfo** result set containing document libraries and lists (1) under the expanded site (2) node.
3. When the client requests information about the document library **Node** under the expanded site (2) node, the front-end Web server gets the information of the document library and its child objects (1) by calling **proc\_GetListAndChildrenNSInfo** (section [3.1.4.21](#)) to the back-end database server.
4. The back-end database server returns one **ChildFoldersNSInfo** result set containing the subfolders under the document library.

### 4.2 HTML Translate Cache

The front-end Web server inserts an item into the HTML translate cache by calling **proc\_HTCreateRow**.

For fetching an item from the HTML translate cache for a document:

1. The front-end Web server first gets the name of the translated file by calling **proc\_HTGuidFromOrig**.
2. It uses the translation name from the Translation Name result set, of the call to **proc\_HTGuidFromOrig**, in the call to **proc\_HTGetFile** to get the item.

### 4.3 Web Discussions Operations

#### 4.3.1 Add a Comment

The front-end Web server adds a comment to a document by calling **proc\_AddDocComment**.

#### 4.3.2 Reply to a Comment

When the user replies to a comment, the front-end Web server calls **proc\_AddDocComment** with the comment identifier that was returned in the **AddDocComment** result set as the *@ParentId* parameter.

#### 4.3.3 Edit a Comment

When the user edits a comment, the front-end Web server calls **proc\_SetCommentAttribs**.

#### 4.3.4 Delete a Comment

Deleting a comment is accomplished by calling **proc\_DeleteDocComment**.

#### 4.3.5 Enumerate All Comments in a Document

When the client requests all the comments of a document, the front-end Web server calls **proc\_GetDocComments** passing the store-relative URL of the document.

### 4.4 Navigation Structure

#### 4.4.1 Create a Navigation Node to a URL

To create a navigation node, the front-end Web server calls **proc\_NavStructAllocateEidBlockWebId**.

The *@EidBase* that is returned is then passed to **proc\_NavStructAddNewNodeByUrl** to create the new navigation node.

#### 4.4.2 Move a Navigation Node

To move a navigation node, the front-end Web server calls **proc\_NavStructAllocateEidBlockWebId**.

The *@EidBase* that is returned is then passed to **proc\_NavStructMoveNode** to move the navigation node.

#### 4.4.3 Enumerate Navigation Nodes in a Site

A front-end Web server enumerates the navigation nodes in a site (2) by calling **proc\_GetWebNavStruct**.

### 4.5 User Resource String

This example describes the requests made by the front-end Web server to add/update translations in different languages for a user resource string.

1. The front-end Web server first gets the existing translations of a user resource string by making a call to **proc\_EnumResourceValuesforAlllangs**.
2. It looks at the result set, from the call to **proc\_EnumResourceValuesforAlllangs**, to determine if a resource needs translation based on the **BitDirty** column (1) in the result set.
3. For resources that need to be updated, it uses the same resource name and calls **proc\_UpdateUserResource** with the corresponding language to add/update the translation of the user resource string.

#### 4.5.1 Add a User Resource String

To add a user resource string, the front-end Web server calls **proc\_AddUserResource**.

#### 4.5.2 Update a User Resource String

To update a user resource string, the front-end Web server calls **proc\_UpdateUserResource**.



## 5 Security

### 5.1 Security Considerations for Implementers

The database access account used by the front-end Web server must have access to the appropriate content database on the back-end database server. If the account does not have the correct access rights, access will be denied when attempting to set up the [\[MS-TDS\]](#) connection to the content database, or when calling the stored procedures.

### 5.2 Index of Security Parameters

This protocol uses the security parameters that are specified in the [\[MS-WSSFO2\]](#) protocol document.

## 6 Appendix A: Product Behavior

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

- Microsoft® SharePoint® Foundation 2010
- Microsoft® SQL Server® 2005
- Microsoft® SQL Server® 2008
- Microsoft® SQL Server® 2008 R2

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

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

[<1> Section 2.2.1.4:](#) Windows SharePoint Services 3.0 treats the values of 2 and 4 in the same manner

[<2> Section 2.2.4.3:](#) Windows SharePoint Services 3.0 does not honor this return value. It hides the Navigation Nodes based on the permissions of the user

[<3> Section 3.1.5.1:](#) If the Web discussion comment corresponding to the @ParentId does not exist, then the server in Windows SharePoint Services v2.0 and v3.0 sets this parameter to zero ("0").

[<4> Section 3.1.5.1.1:](#) Windows SharePoint Services v2.0 and v3.0 tries to add the Web discussion comment 10 times with a different value for **Id** each time. If it was successful it returns zero and the result set contains the successfully inserted values. If the server fails to add the comment, the server returns an error code other than zero and the result set contains the values that it tried to insert in the final attempt.

[<5> Section 3.1.5.30.3:](#) Windows SharePoint Services 3.0 does not honor this return value. It hides the Navigation Nodes based on the permissions of the user

[<6> Section 3.1.5.36:](#) The client in Windows SharePoint Services 3.0 creates a unique Translation Name for each file. Thus there is only one row (3) corresponding to a SiteID, DirName, LeafName in the table.

[<7> Section 3.1.5.44:](#) Windows SharePoint Services 3.0 does not honor this parameter. It hides the Navigation Nodes based on the permissions of the user.

## 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
  - [server](#) 21
- [Add a comment example](#) 119
- [Add a user resource string example](#) 120
- [AddAndUpdateNavigationNode sequencing rules](#) 117
- Alert Delivery Channel simple type ([section 2.2.1.1](#) 14, [section 2.2.1.1](#) 14)
- [Alert event type bit field](#) 16
- Alert simple type ([section 2.2.1.3](#) 14, [section 2.2.1.3](#) 14)
- Alert Status simple type ([section 2.2.1.2](#) 14, [section 2.2.1.2](#) 14)
- Alerts
  - [server](#) 21
  - [Alerts overview](#) 11
  - [AllUserData table structure](#) 20
  - [Applicability](#) 13
  - [Attribute groups - overview](#) 20
  - [Attributes - overview](#) 20

### B

- Binary structures
  - [Cached Nav Data](#) 17
- [Binary structures - overview](#) 17
- Bit fields
  - [alert event type](#) 16

### C

- [Cached Nav Data binary structure](#) 17
- [Capability negotiation](#) 13
- [Change tracking](#) 123
- Client
  - [higher-layer triggered events](#) 115
  - [initialization](#) 115
  - [local events](#) 118
  - [message processing](#) 118
  - [overview](#) 114
  - [sequencing rules](#) 118
  - [timer events](#) 118
  - [timers](#) 114
  - [WSSEUX interface](#) 114
- Client - sequencing rules
  - [AddAndUpdateNavigationNode](#) 117
  - [CreateNavigationNodeByDocId](#) 116
  - [CreateNavigationNodeByUrl](#) ([section 3.2.4.3](#) 116, [section 3.2.4.3](#) 116)
  - [fetch HTML translate cache item](#) 115
  - [MoveNavigationNode](#) 117
- Common data types
  - [overview](#) 14
- [Complex types - overview](#) 20
- [Create a navigation node to a URL example](#) 120
- [CreateNavigationNodeByDocId sequencing rules](#) 116

### D

- Data model - abstract
  - [server](#) 21
- Data types
  - Alert Delivery Channel simple type ([section 2.2.1.1](#) 14, [section 2.2.1.1](#) 14)
  - Alert simple type ([section 2.2.1.3](#) 14, [section 2.2.1.3](#) 14)
  - Alert Status simple type ([section 2.2.1.2](#) 14, [section 2.2.1.2](#) 14)
  - [common](#) 14
  - Meetings Event simple type ([section 2.2.1.4](#) 15, [section 2.2.1.4](#) 15)
  - Navigation Node simple type ([section 2.2.1.5](#) 15, [section 2.2.1.5](#) 15)
  - Notification Frequency simple type ([section 2.2.1.6](#) 15, [section 2.2.1.6](#) 15)
  - RecurrenceId simple type ([section 2.2.1.7](#) 15, [section 2.2.1.7](#) 15)
  - Sequence simple type ([section 2.2.1.8](#) 15, [section 2.2.1.8](#) 15)
  - UID simple type ([section 2.2.1.9](#) 16, [section 2.2.1.9](#) 16)
  - User Resource simple type ([section 2.2.1.10](#) 16, [section 2.2.1.10](#) 16)
- Data types - simple
  - [Alert](#) 14
  - [Alert Delivery Channel](#) 14
  - [Alert Status](#) 14
  - [Meetings Event](#) 15
  - [Navigation Node](#) 15
  - [Notification Frequency](#) 15
  - [overview](#) 14
  - [RecurrenceId](#) 15
  - [Sequence](#) 15
  - [UID](#) 16
  - [User Resource](#) 16
- Data types – simple
  - [alert](#) 14
  - [alert delivery channel](#) 14
  - [alert status](#) 14
  - [meetings event](#) 15
  - [navigation node](#) 15
  - [notification frequency](#) 15
  - [RecurrenceId](#) 15
  - [sequence](#) 15
  - [UID](#) 16
  - [user resource](#) 16
- [Delete a comment example](#) 120

### E

- [Edit a comment example](#) 119
- [Elements - overview](#) 20
- [E-mail integration overview](#) 11
- [EmailAliases result set](#) 18

[Enumerate all comments in a document example](#)  
120

[Enumerate navigation nodes in a site example](#) 120

#### Events

[local - client](#) 118

[local - server](#) 114

[timer - client](#) 118

[timer - server](#) 114

#### Examples

[add a comment](#) 119

[add a user resource string](#) 120

[create a navigation node to a URL](#) 120

[delete a comment](#) 120

[edit a comment](#) 119

[enumerate all comments in a document](#) 120

[enumerate navigation nodes in a site](#) 120

[HTML translate cache](#) 119

[move a navigation node](#) 120

[reply to a comment](#) 119

[TreeView operations](#) 119

[update a user resource string](#) 120

[user resource string](#) 120

## F

[Fetch HTML translate cache item sequencing rules](#)  
115

[Fields - vendor-extensible](#) 13

#### Flag structures

[special alert](#) 16

[Web discussion comment status](#) 17

## G

[Glossary](#) 8

[Groups - overview](#) 20

## H

#### Higher-layer triggered events

[client](#) 115

[server](#) 24

#### HTML translate cache

[server](#) 21

[HTML translate cache example](#) 119

[HTML translate cache overview](#) 12

## I

[Implementer - security considerations](#) 121

[Index of security parameters](#) 121

[Informative references](#) 11

#### Initialization

[client](#) 115

[server](#) 24

#### Interfaces - client

[WSSEUX](#) 114

[Introduction](#) 8

## L

#### Local events

[client](#) 118

[server](#) 114

## M

#### Meetings

[server](#) 22

Meetings Event simple type ([section 2.2.1.4](#) 15,

[section 2.2.1.4](#) 15)

[Meetings overview](#) 12

#### Message processing

[client](#) 118

[server](#) 24

#### Messages

[alert event type bit field](#) 16

[AllUserData table structure](#) 20

[attribute groups](#) 20

[attributes](#) 20

[binary structures](#) 17

[Cached Nav Data binary structure](#) 17

[common data types](#) 14

[complex types](#) 20

[elements](#) 20

[EmailAliases result set](#) 18

[enumerations](#) 14

[groups](#) 20

[namespaces](#) 20

[Nav ACLs result set](#) 18

[Nav Children result set](#) 20

[Nav Data result set](#) 19

[simple data types](#) 14

[simple types](#) 20

[special alert flag structure](#) 16

[transport](#) 14

[UserInfo table structure](#) 20

[Web discussion comment status flag structure](#) 17

[WebId result set](#) 105

[XML structures](#) 20

#### Methods

[proc\\_AddDocComment](#) 24

[proc\\_AddSubscription](#) 26

[proc\\_AddUserResource](#) 29

[proc\\_CheckMeetingInstance](#) 30

[proc\\_CheckNavStructContainsPage](#) 32

[proc\\_DeleteSubscription](#) 33

[proc\\_DeleteSubscriptionJunctionEntries](#) 33

[proc\\_EnumDoclibsFileDlg](#) 38

[proc\\_EnumEmailAliases](#) 40

[proc\\_EnumEmailAliasesBySite](#) 40

[proc\\_EnumResourcesAtScope](#) 40

[proc\\_EnumResourceValuesForAllLangs](#) 41

[proc\\_EnumSubscribedSites](#) 42

[proc\\_GetAlertsSqmData](#) 43

[proc\\_GetDefaultMtgInstance](#) 43

[proc\\_GetDocComments](#) 45

[proc\\_GetEventDataAndSubscriptionFilters](#) 46

[proc\\_GetExceptionIDs](#) 49

[proc\\_GetFutureExceptionIDsForUID](#) 50

[proc\\_GetItemCountPerInstance](#) 52

[proc\\_GetListAndChildrenNSInfo](#) 52

[proc\\_GetMatchingSubscriptionsData](#) 55

[proc\\_GetMeetingInstanceDataForICal](#) 61

- [proc\\_GetRecurrenceSeriesData](#) 62
- [proc\\_GetResourcesAtScope](#) 64
- [proc\\_GetVersionIndependentMetaInfo](#) 65
- [proc\\_GetWebAndChildrenNSInfo](#) 66
- [proc\\_GetWebComments](#) 69
- [proc\\_GetWebNavAcls](#) 70
- [proc\\_GetWebNavStruct](#) 71
- [proc\\_GetWebNavStructNodeByIds](#) 73
- [proc\\_GetWebSubscriptions](#) 73
- [proc\\_GetWebSubscriptionsForBackup](#) 75
- [proc\\_GetWebSubscriptionsUniqueUsers](#) 77
- [proc\\_HandleMtgRecurPatternChange](#) 77
- [proc\\_HTCreateRow](#) 78
- [proc\\_HTGetFile](#) 79
- [proc\\_HTGuidFromOrig](#) 80
- [proc\\_InstantiateMtgSeriesOccurrence](#) 81
- [proc\\_ListDocsHavingComments](#) 83
- [proc\\_MatchSchedSubscriptions](#) 84
- [proc\\_ModifySubscription](#) 87
- [proc\\_MoveDataToOccurrence](#) 89
- [proc\\_NavStructAddNewNodeByDocId](#) 91
- [proc\\_NavStructAddNewNodeByUrl](#) 93
- [proc\\_NavStructAllocateEidBlockWebId](#) 95
- [proc\\_NavStructClear](#) 96
- [proc\\_NavStructDeleteNodeByIds](#) 96
- [proc\\_NavStructLogChangesAndUpdateSiteChange](#)  
[dTime](#) 97
- [proc\\_NavStructMoveNode](#) 97
- [proc\\_OrphanRecurringEventExceptions](#) 99
- [proc\\_ProcessDelMtgAttendeeListItem](#) 100
- [proc\\_PutWebNavStructNode](#) 101
- [proc\\_RelinkMeeting](#) 104
- [proc\\_ResolveWikiLinkFile](#) 104
- [proc\\_ResolveWikiLinkItem](#) 105
- [proc\\_SetCommentAttribs](#) 106
- [proc\\_SetSecondaryResourceAtScope](#) 107
- [proc\\_SetVersionIndependentMetaInfo](#) 107
- [proc\\_SetWebNavStructInheritance](#) 108
- [proc\\_UnOrphanMtgOccurrence](#) 109
- [proc\\_UpdateCachedNav](#) 110
- [proc\\_UpdateItemJunctionsCurrentVersion](#) 111
- [proc\\_UpdateListNavNode](#) 112
- [proc\\_UpdateSchedSubscriptionTimes](#) 112
- [proc\\_UpdateUserResource](#) 113
- [Move a navigation node example](#) 120
- [MoveNavigationNode sequencing rules](#) 117
- Multilingual user interface
  - [server](#) 22
- [Multilingual user interface overview](#) 12

**N**

- [Namespaces](#) 20
- [Nav ACLs result set](#) 18
- [Nav Children result set](#) 20
- [Nav Data result set](#) 19
- Navigation Node simple type ([section 2.2.1.5](#) 15,  
[section 2.2.1.5](#) 15)
- Navigation structure
  - [server](#) 23
- [Navigation structure overview](#) 12
- [Normative references](#) 10

Notification Frequency simple type ([section 2.2.1.6](#)  
15, [section 2.2.1.6](#) 15)

**O**

[Overview \(synopsis\)](#) 11

**P**

- [Parameters - security index](#) 121
- [Preconditions](#) 13
- [Prerequisites](#) 13
- [proc\\_AddDocComment method](#) 24
- [proc\\_AddSubscription method](#) 26
- [proc\\_AddUserResource method](#) 29
- [proc\\_CheckMeetingInstance method](#) 30
- [proc\\_CheckNavStructContainsPage method](#) 32
- [proc\\_DeleteSubscription method](#) 33
- [proc\\_DeleteSubscriptionJunctionEntries method](#) 33
- [proc\\_EnumDoclibsFileDlq method](#) 38
- [proc\\_EnumEmailAliases method](#) 40
- [proc\\_EnumEmailAliasesBySite method](#) 40
- [proc\\_EnumResourcesAtScope method](#) 40
- [proc\\_EnumResourceValuesForAllLangs method](#) 41
- [proc\\_EnumSubscribedSites method](#) 42
- [proc\\_GetAlertsSqmData method](#) 43
- [proc\\_GetDefaultMtgInstance method](#) 43
- [proc\\_GetDocComments method](#) 45
- [proc\\_GetEventDataAndSubscriptionFilters method](#)  
46
- [proc\\_GetExceptionIDs method](#) 49
- [proc\\_GetFutureExceptionIDsForUID method](#) 50
- [proc\\_GetItemCountPerInstance method](#) 52
- [proc\\_GetListAndChildrenNSInfo method](#) 52
- [proc\\_GetMatchingSubscriptionsData method](#) 55
- [proc\\_GetMeetingInstanceDataForICal method](#) 61
- [proc\\_GetRecurrenceSeriesData method](#) 62
- [proc\\_GetResourcesAtScope method](#) 64
- [proc\\_GetVersionIndependentMetaInfo method](#) 65
- [proc\\_GetWebAndChildrenNSInfo method](#) 66
- [proc\\_GetWebComments method](#) 69
- [proc\\_GetWebNavAcls method](#) 70
- [proc\\_GetWebNavStruct method](#) 71
- [proc\\_GetWebNavStructNodeByIds method](#) 73
- [proc\\_GetWebSubscriptions method](#) 73
- [proc\\_GetWebSubscriptionsForBackup method](#) 75
- [proc\\_GetWebSubscriptionsUniqueUsers method](#) 77
- [proc\\_HandleMtgRecurPatternChange method](#) 77
- [proc\\_HTCreateRow method](#) 78
- [proc\\_HTGetFile method](#) 79
- [proc\\_HTGuidFromOrig method](#) 80
- [proc\\_InstantiateMtgSeriesOccurrence method](#) 81
- [proc\\_ListDocsHavingComments method](#) 83
- [proc\\_MatchSchedSubscriptions method](#) 84
- [proc\\_ModifySubscription method](#) 87
- [proc\\_MoveDataToOccurrence method](#) 89
- [proc\\_NavStructAddNewNodeByDocId method](#) 91
- [proc\\_NavStructAddNewNodeByUrl method](#) 93
- [proc\\_NavStructAllocateEidBlockWebId method](#) 95
- [proc\\_NavStructClear method](#) 96
- [proc\\_NavStructDeleteNodeByIds method](#) 96

[proc\\_NavStructLogChangesAndUpdateSiteChangedTime method](#) 97  
[proc\\_NavStructMoveNode method](#) 97  
[proc\\_OrphanRecurringEventExceptions method](#) 99  
[proc\\_ProcessDelMtgAttendeeListItem method](#) 100  
[proc\\_PutWebNavStructNode method](#) 101  
[proc\\_RelinkMeeting method](#) 104  
[proc\\_ResolveWikiLinkFile method](#) 104  
[proc\\_ResolveWikiLinkItem method](#) 105  
[proc\\_SetCommentAttribs method](#) 106  
[proc\\_SetSecondaryResourceAtScope method](#) 107  
[proc\\_SetVersionIndependentMetaInfo method](#) 107  
[proc\\_SetWebNavStructInheritance method](#) 108  
[proc\\_UnOrphanMtgOccurrence method](#) 109  
[proc\\_UpdateCachedNav method](#) 110  
[proc\\_UpdateItemJunctionsCurrentVersion method](#) 111  
[proc\\_UpdateListNavNode method](#) 112  
[proc\\_UpdateSchedSubscriptionTimes method](#) 112  
[proc\\_UpdateUserResource method](#) 113  
[Product behavior](#) 122

## R

[RecurrenceId simple type \(section 2.2.1.7 15, section 2.2.1.7 15\)](#)

[References](#) 10

[informative](#) 11

[normative](#) 10

[Relationship to other protocols](#) 12

[Reply to a comment example](#) 119

[Result sets - messages](#)

[EmailAliases](#) 18

[Nav ACLs](#) 18

[Nav Children](#) 20

[Nav Data](#) 19

[WebId](#) 105

## S

[Security](#)

[implementer considerations](#) 121

[parameter index](#) 121

[Sequence simple type \(section 2.2.1.8 15, section 2.2.1.8 15\)](#)

[Sequencing rules](#)

[client](#) 118

[server](#) 24

[Sequencing rules - client](#)

[AddAndUpdateNavigationNode](#) 117

[CreateNavigationNodeByDocId](#) 116

[CreateNavigationNodeByUrl sequencing rules](#) 116

[fetch HTML translate cache item](#) 115

[MoveNavigationNode](#) 117

[Server](#)

[abstract data model](#) 21

[higher-layer triggered events](#) 24

[HTML translate cache](#) 21

[initialization](#) 24

[local events](#) 114

[meetings](#) 22

[message processing](#) 24

[multilingual user interface](#) 22

[navigation structure](#) 23

[proc\\_AddDocComment method](#) 24

[proc\\_AddSubscription method](#) 26

[proc\\_AddUserResource method](#) 29

[proc\\_CheckMeetingInstance method](#) 30

[proc\\_CheckNavStructContainsPage method](#) 32

[proc\\_DeleteSubscription method](#) 33

[proc\\_DeleteSubscriptionJunctionEntries method](#)

33

[proc\\_EnumDoclibsFileDlg method](#) 38

[proc\\_EnumEmailAliases method](#) 40

[proc\\_EnumEmailAliasesBySite method](#) 40

[proc\\_EnumResourcesAtScope method](#) 40

[proc\\_EnumResourceValuesForAllLangs method](#) 41

[proc\\_EnumSubscribedSites method](#) 42

[proc\\_GetAlertsSqmData method](#) 43

[proc\\_GetDefaultMtgInstance method](#) 43

[proc\\_GetDocComments method](#) 45

[proc\\_GetEventDataAndSubscriptionFilters method](#)

46

[proc\\_GetExceptionIDs method](#) 49

[proc\\_GetFutureExceptionIDsForUID method](#) 50

[proc\\_GetItemCountPerInstance method](#) 52

[proc\\_GetListAndChildrenNSInfo method](#) 52

[proc\\_GetMatchingSubscriptionsData method](#) 55

[proc\\_GetMeetingInstanceDataForICal method](#) 61

[proc\\_GetRecurrenceSeriesData method](#) 62

[proc\\_GetResourcesAtScope method](#) 64

[proc\\_GetVersionIndependentMetaInfo method](#) 65

[proc\\_GetWebAndChildrenNSInfo method](#) 66

[proc\\_GetWebComments method](#) 69

[proc\\_GetWebNavAcls method](#) 70

[proc\\_GetWebNavStruct method](#) 71

[proc\\_GetWebNavStructNodeByIds method](#) 73

[proc\\_GetWebSubscriptions method](#) 73

[proc\\_GetWebSubscriptionsForBackup method](#) 75

[proc\\_GetWebSubscriptionsUniqueUsers method](#)

77

[proc\\_HandleMtgRecurPatternChange method](#) 77

[proc\\_HTCreateRow method](#) 78

[proc\\_HTGetFile method](#) 79

[proc\\_HTGuidFromOrig method](#) 80

[proc\\_InstantiateMtgSeriesOccurrence method](#) 81

[proc\\_ListDocsHavingComments method](#) 83

[proc\\_MatchSchedSubscriptions method](#) 84

[proc\\_ModifySubscription method](#) 87

[proc\\_MoveDataToOccurrence method](#) 89

[proc\\_NavStructAddNewNodeByDocId method](#) 91

[proc\\_NavStructAddNewNodeByUrl method](#) 93

[proc\\_NavStructAllocateEidBlockWebId method](#) 95

[proc\\_NavStructClear method](#) 96

[proc\\_NavStructDeleteNodeByIds method](#) 96

[proc\\_NavStructLogChangesAndUpdateSiteChange](#)

dTime method 97

[proc\\_NavStructMoveNode method](#) 97

[proc\\_OrphanRecurringEventExceptions method](#)

99

[proc\\_ProcessDelMtgAttendeeListItem method](#) 100

[proc\\_PutWebNavStructNode method](#) 101

[proc\\_RelinkMeeting method](#) 104

- [proc ResolveWikiLinkFile method](#) 104
- [proc ResolveWikiLinkItem method](#) 105
- [proc SetCommentAttribs method](#) 106
- [proc SetSecondaryResourceAtScope method](#) 107
- [proc SetVersionIndependentMetaInfo method](#) 107
- [proc SetWebNavStructInheritance method](#) 108
- [proc UnOrphanMtgOccurrence method](#) 109
- [proc UpdateCachedNav method](#) 110
- [proc UpdateItemJunctionsCurrentVersion method](#) 111
- [proc UpdatelistNavNode method](#) 112
- [proc UpdateSchedSubscriptionTimes method](#) 112
- [proc UpdateUserResource method](#) 113
- [sequencing rules](#) 24
- [timer events](#) 114
- [timers](#) 24
- [TreeView](#) 24
- [Web discussion](#) 24
- Simple data types
  - [Alert](#) 14
  - Alert Delivery Channel ([section 2.2.1.1](#) 14, [section 2.2.1.1](#) 14)
  - Alert Status ([section 2.2.1.2](#) 14, [section 2.2.1.2](#) 14)
  - [Meetings Event](#) 15
  - [Navigation Node](#) 15
  - [Notification Frequency](#) 15
  - [overview](#) 14
  - [RecurrenceId](#) 15
  - [Sequence](#) 15
  - [UID](#) 16
  - [User Resource](#) 16
- Simple types
  - [alert](#) 14
  - [meetings event](#) 15
  - [navigation node](#) 15
  - [notification frequency](#) 15
  - [RecurrenceId](#) 15
  - [sequence](#) 15
  - [UID](#) 16
  - [user resource](#) 16
- [Simple types - overview](#) 20
- [Special alert flag structure](#) 16
- [Standards assignments](#) 13
- Structures
  - [binary](#) 17
  - [XML](#) 20

## T

- Table structures
  - [AllUserData](#) 20
  - [UserInfo](#) 20
- Timer events
  - [client](#) 118
  - [server](#) 114
- Timers
  - [client](#) 114
  - [server](#) 24
- [Tracking changes](#) 123
- [Transport](#) 14

- [Tree view overview](#) 12
- TreeView
  - [server](#) 24
- [TreeView operations example](#) 119
- Triggered events - higher-layer
  - [client](#) 115
  - [server](#) 24
- Types
  - [complex](#) 20
  - [simple](#) 20

## U

- UID simple type ([section 2.2.1.9](#) 16, [section 2.2.1.9](#) 16)
- [Update a user resource string example](#) 120
- User Resource simple type ([section 2.2.1.10](#) 16, [section 2.2.1.10](#) 16)
- [User resource string example](#) 120
- [UserInfo table structure](#) 20

## V

- [Vendor-extensible fields](#) 13
- [Versioning](#) 13

## W

- Web discussion
  - [server](#) 24
- [Web discussion comment status flag structure](#) 17
- [Web discussions overview](#) 12
- [WebId result set](#) 105
- [WSSEUX interface](#) 114

## X

- [XML structures](#) 20