[MS-WSSEUX]:
Windows SharePoint Services:
Content Database End-User Experience Communications Protocol Specification

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.

- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.

- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.

- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft's Open Specification Promise (available here: http://www.microsoft.com/interop/osp) or the Community Promise (available here: http://www.microsoft.com/interop/cp/default.mspx). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting 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
04/04/2008	0.1		Initial Availability
04/25/2008	0.2	Editorial	Revised and edited the technical content
06/27/2008	1.0	Major	Revised and edited the technical content
10/06/2008	1.01	Editorial	Revised and edited the technical content
12/12/2008	1.02	Editorial	Revised and edited the technical content
03/18/2009	1.03	Editorial	Revised and edited the technical content
07/13/2009	1.04	Major	Changes made for template compliance
08/28/2009	1.05	Editorial	Revised and edited the technical content
11/06/2009	1.06	Editorial	Revised and edited the technical content
02/19/2010	2.0	Minor	Updated the technical content
03/31/2010	2.01	Editorial	Revised and edited the technical content
04/30/2010	2.02	Editorial	Revised and edited the technical content
06/07/2010	2.03	Editorial	Revised and edited the technical content
06/29/2010	2.04	Editorial	Changed language and formatting in the technical content.
07/23/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
09/27/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	2.05	Editorial	Changed language and formatting in the technical content.

Table of Contents

Release: Sunday, December 19, 2010

1 Introduction

This document specifies the Windows SharePoint Services: Content Database End-User Experience Communications 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.

1.1 Glossary

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

access control list (ACL)
anonymous user
globally unique identifier (GUID)
language code identifier (LCID)
unique identifier (UID)
UTC (Coordinated Universal Time)
XML

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

12-hour clock notation
24-hour clock notation
alert
alert subscription
always notify alert
back-end database server
calendar type
Central Administration site
change log
checked out
Collaborative Application Markup Language (CAML)
collation order
column
content database
current user
datetime
default list view
deleted
directory name
display name
document
document library
document stream
e-mail alias
embedded image
event
field
file
folder
front-end Web server
home page
HTML translate cache
iCalendar
inheritable navigation structure

item identifier
leaf name
link bar
list
list identifier
list item
list view page
locale
locked
login name
meeting instance
Meeting Workspace site
metadict
navigation node
navigation node element identifier
navigation structure
navigational parent site
permission
published
query
read-only mode
result set
return code
root folder
row
rule
scope identifier
security principal
security scope
sequence number
server-relative URL
site
site collection
site collection identifier
site definition
site identifier
site template
site-relative URL
SQL collation name
stored procedure
store-relative URL
subsite
system alert
T-SQL (Transact-Structured Query Language)
URL (Uniform Resource Locator)
user identifier
user name
Web discussion comment
Windows collation name

The following terms are specific to this document:

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

1.2 References

1.2.1 Normative References

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

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

[MS-SQL] Microsoft Corporation, "SQL Server 2000 Architecture and XML/Internet Support", Volume 1 of Microsoft SQL Server 2000 Reference Library, Microsoft Press, 2001, ISBN 0-7356-1280-3, http://msdn.microsoft.com/en-us/library/dd631854(v=SQL.10).aspx

[MS-TDS] Microsoft Corporation, "Tabular Data Stream Protocol Specification", February 2008.

[MS-WSSFO] Microsoft Corporation, "Windows SharePoint Services (WSS): File Operations Database Communications Protocol Specification", April 2008.

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

[RFC2445] Dawson, F. and Stenerson, D., "Internet Calendaring and Scheduling Core Object Specification (iCalendar)", RFC 2445, November 1998, http://tools.ietf.org/html/rfc2445

1.2.2 Informative References

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary", March 2007.

[MS-OFCGLOS] Microsoft Corporation, "Microsoft Office Master Glossary", June 2008.

[MS-OUTSPS] Microsoft Corporation, "Lists Client Sync Protocol Specification", June 2008.

[MS-WSSDLIM] Microsoft Corporation, "Windows SharePoint Services: Content Database Document and List Item Management Communications Protocol Specification", June 2008.

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 and sites. 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 SharePoint object) 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 for a user, **site** or **list item** or modify the alerts. 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 with documents (or other SharePoint objects) 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 Autocopy

The front-end Web server provides a mechanism for the user to copy documents contained in a particular **folder** or **list** to a different location. If the document is copied to the same location as the source, a duplicate copy with a different name is made. The destination folder or list could be in a different site from the source as well. Both the content of the document (stored in the **document stream**) and the metadata associated with the document are copied.

1.3.3 E-mail Integration

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

1.3.4 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**.

1.3.5 Meetings

Users can create calendar events and meetings in a site. They are created as list items in an events list. Users can enumerate and update these items with new information. They can associate a **Meeting Workspace site** with each meeting. A meeting workspace 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 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 15th of each month when it occurs at 11 A.M.

1.3.6 Navigation Structure

The **navigation structure** is a hierarchical representation of related URLs. Every site MUST have 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 can contain a number of subsites, lists, 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. 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, 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:
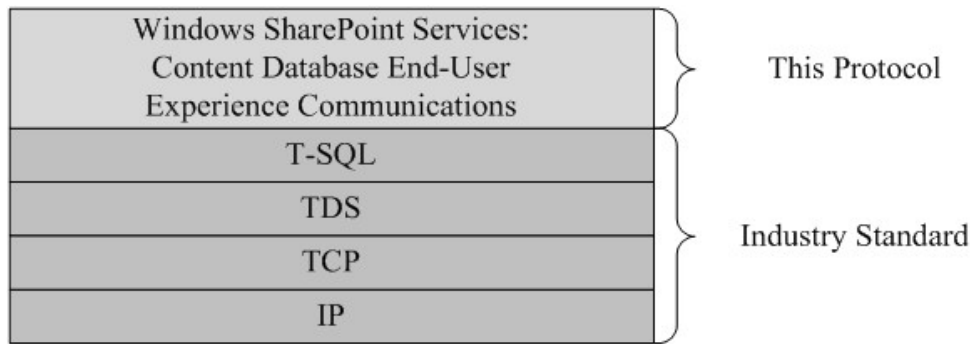
Figure 1: This protocol in relation to other protocols

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

2.2.2   Simple Data Types

2.2.2.1   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

2.2.2.2   Alert

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

Value	Description
0	List alert
1	Item alert
2	Custom alert

2.2.2.3   Meetings Event

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

Value	Description
0	A single nonrecurring event.
1	A recurring event
2	An exception from the recurrence series.
3	An exception that has been deleted.

Value	Description
4	An exception from the recurrence series<1>.
5	A single instance of a meetings workspace.

2.2.2.4 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.

2.2.2.5 Notification Frequency

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

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

2.2.2.6 RecurrenceId

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

2.2.2.7 Sequence

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

2.2.2.8 UID

This element contains a unique identifier specifying a meeting. This MUST be a valid **unique identifier (UID)** string as used by calendaring objects 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.

2.2.3   Bit Fields and Flag Structures

2.2.3.1   Alert Event Type

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

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

2.2.3.2   Special Alert Flags

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

Value	Description
0x40000000	Indicates that the alert is an always notify alert.
0x20000000	Indicates that the alert is a system alert.

2.2.4   Enumerations

None.

2.2.5   Binary Structures

None.

2.2.6   Result Sets

2.2.6.1   EmailAliases Result Set

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 to which the e-mail alias belongs.

tp_ID: The identifier of the list to which the e-mail alias belongs.

2.2.6.2 Nav ACLs Result Set

Returns the ACLs of all the unique security scopes of documents in the navigation structure of the site. This result set MUST be returned. The number of rows in this result set depends on the values of *@NavParentWebId* and *@Inherited*.

Value of @NavParentWebId	Value of @Inherited	Number of rows returned
NULL	0	One row for each distinct **security scope** for the navigation nodes of the site that are other than the security scope of the site. If there are no distinct security scopes for the navigation nodes of the site other than the security scope, then this result set MUST return 0 rows.
NULL	1	One row for each distinct security scope for the navigation nodes of the site.
NOT NULL	0/1	One row for each distinct security scope for the navigation nodes of the site, one row for each distinct security scope for the navigation nodes of the **navigational parent site**, and one row for the security scope of the navigational parent site.

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

```
ScopeId               uniqueidentifier,
Acl                   image,
AnonymousPermMask     bigint;
```

ScopeId: The identifier of the security scope.

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

AnonymousPermMask: The WSS Rights Mask, as described in [MS-WSSFO] section 2.2.2.13, that indicates the rights granted to an **anonymous user**, or a user who has no specific rights for this security scope.

2.2.6.3 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, this MUST return one row; otherwise it MUST return no rows. The T-SQL syntax for the result set is as follows:

```
Eid                   int,
EidParent             int,
ElementType           tinyint,
{Url}                 nvarchar(260),
Name                  nvarchar(256),
NodeMetainfo          image,
NonNavPage            bit,
```

```
NavSequence          bit,
DateLastModified     datetime;
```

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

EidParent: The identifier of the parent navigation node.

ElementType: This represents the Navigation Node Type (Section 2.2.2.4) 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 0.

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

NodeMetainfo: A binary serialization of the navigation node **metadata**.

NonNavPage: A bit representing whether the navigation node will be filtered out when rendering the navigation structure. If the bit is set to 1, then the navigation node SHOULD be filtered out<2>. If it is set to 0, it MUST NOT be filtered out.

NavSequence: A bit representing 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 0.

ChildOfSequence: A bit representing whether the navigation node is a child navigation node of a link bar. If it does, this bit MUST be set to 1; otherwise it MUST be set to 0

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

2.2.6.4 Nav Children Result Set

Returns the navigation node element identifier of all the subnodes of the navigation node. This result set MUST be returned, and it MUST contain one row for each navigation node that is a child of the specified navigation node. If the navigation node has no subnodes, this MUST return no rows. The T-SQL syntax for the result set is as follows:

```
Eid                  int;
```

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

2.2.7 Tables and Views

2.2.7.1 AllUserData Table

Specified in [MS-WSSFO] section 2.2.6.3.

2.2.7.2 UserInfo Table

Specified in [MS-WSSFO] section 2.2.7.10.

2.2.8 XML Structures

None.

2.2.8.1 Namespaces

None.

2.2.8.2 Simple Types

None.

2.2.8.3 Complex Types

None.

2.2.8.4 Elements

None.

2.2.8.5 Attributes

None.

2.2.8.6 Groups

None.

2.2.8.7 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 of result sets imply the kinds and types of data that MUST be stored by the front-end Web servers and back-end database servers. In particular, note the following:

- The back-end database server MUST store, for each site, an integer 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 (alerts) that a particular document or list has been modified. The user can also request notifications only for certain kinds of modifications. For example, an alert can be requested if the value of a specific field 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 subscription, then it sends an e-mail notification to the user.

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

The changes to the objects are stored as events in the event cache table. (For information about event cache table, see [MS-WSSDLIM].) The server matches the events in the event cache with the list of immediate and scheduled subscriptions and sends notifications for events that match 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 3.1.4.36). 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 (Section 3.1.4.36), 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 which performed the translation to HTML, and all links to supporting files within the main file are relative links of the form <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 previous 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 (Section 3.1.4.36) and proc_HTGetFile (Section 3.1.4.35) 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. They are stored as list items inside a list which 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 which is expressed in **XML (Extensible Markup Language)** 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, 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-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 can be associated with multiple single instance meetings or one recurring meeting.

3.1.1.4 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 MAY also have metadata associated with it.

When adding navigation nodes, temporary navigation node element identifiers MAY be used. navigation node element identifiers 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 navigation node element identifiers MUST contain a base value. This MUST be obtained by a call to proc_NavStructAllocateEidBlockWebId (Section 3.1.4.44). The base value MUST be used to generate valid navigation node element identifiers from the temporary navigation node element identifiers.

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 3.1.4.44) 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. If a new navigation node pointing to the home page of the site and of navigation node Type 0 is being added NOT 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 3.1.4.42) and proc_NavStructAddNewNodeByUrl (Section 3.1.4.43). This parameter MUST be used by all following SQL queries in the batch, as the *@EidHome* parameter for proc_NavStructMoveNode (Section 3.1.4.48) and proc_PutWebNavStructNode (Section 3.1.4.51) and as the *@@EidHome* parameter for proc_NavStructAddNewNodeByDocId (Section 3.1.4.42) and proc_NavStructAddNewNodeByUrl (Section 3.1.4.43).

A part of the navigation structure of a site 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 MAY be cached for faster access.

3.1.1.5 TreeView

Data in the site is stored in a hierarchical manner. A site can contain subsites, lists, and **document libraries**. Lists 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.

3.1.1.6 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 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 which 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** 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 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 {*name*}.

3.1.4.1 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          ntext,
    @Size             int,
    @UserId           int,
    @UserTitle        nvarchar(255),
    @Created          datetime
);
```

@SiteId: The identifier of 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 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 flag value that indicates the status of a Web discussion comment.

Release: Sunday, December 19, 2010

@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.

Return Values: proc_AddDocComment returns an integer **return code** which MUST be listed in the following table:

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

Result Sets: The stored procedure MUST return 0 or 1 result sets as follows:

3.1.4.1.1 AddDocComment Result Set

Returns information about the Web discussion comment that was added. If the return code is 0, the Add Doc Comment result set MUST be returned and MUST contain one row. 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 result set is defined using T-SQL syntax, 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.

3.1.4.2 proc_AddSubscription

The **proc_AddSubscription** stored procedure is called to add an alert. A list level alert is an alert which gets triggered for changes to all the list items in a list or when new list items are added to the list. An item level alert is an alert that is triggered for changes to a single list item. 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(64),
        @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              ntext,
        @ItemDocUrl              nvarchar(260),
        @ItemDocId               uniqueidentifier,
        @Max                     int,
        @NewSubId                uniqueidentifier = NULL OUTPUT,
        @UserEmail               nvarchar(255) = NULL OUTPUT,
        @ItemName                nvarchar(255) = NULL OUTPUT
    );
```

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

@NotifyFreq: A Notification Frequency Type (Section 2.2.2.5) that specifies an integer indicating the frequency of the alert notification.

@NotifyTime: The time at which the alert is supposed to fire. This MUST NOT be NULL.

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

@Status: An Alert Status Type (Section 2.2.2.1) that specifies an integer indicating the status of the alert subscription.

@WebId: The GUID of the site in which the list or item exist. This MUST NOT be NULL.

@ListId: The GUID of the list for a list level alert or the list containing the list item for an item level alert. This MUST NOT be NULL.

@ItemId: The identifier of the list item for an item level alert. This MUST NOT be NULL if the Alert Type (Section 2.2.2.2) of the subscription is item and MUST be NULL if the Alert Type (Section 2.2.2.2) of the subscription is list.

@EventType: An Alert Event Type (Section 2.2.3.1) that specifies an integer mask for the types of events on which to fire the notification.

@UserId: The identifier of the user as described in userInfo table for whom the alert will be created. This MUST NOT be NULL.

@SiteUrl: The URL of the site collection in which the list exists.

@WebUrl: The URL of the site in which the list exists.

@WebTitle: The title of the site in which the list or item exists.

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

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

@WebTimeZone: An integer representing the Time Zone of the site. 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 format for the site. This MUST NOT be NULL.

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

@ListUrl: The URL of the list.

@ListTitle: The title of the list.

@ListBaseType: The base type of the list with which this subscription is associated. See [MS-WSSFO] section 2.2.2.13.

@ListServerTemplate: The server template of the list. See [MS-WSSFO] section 2.2.3.10 for details.

@AlertTitle: The title of the alert for display in UI.

@AlertType: An integer whose lower 8 bits specifies the type of the alert as specified in Alert Type (Section 2.2.2.2). This MUST also contain the Special Alert Flags (Section 2.2.3.2) for an always notify alert or a system alert. This MUST NOT be NULL.

@AlertTemplateName: The name of the alert template which will be used for formatting the notification.

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

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

@Properties: An XML blob representing the properties of the alert.

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

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

@Max: The maximum number of alerts allowed in a content database.

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

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

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

Return Values: The stored procedure returns an integer return code which 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.

3.1.4.3   proc_CheckMeetingInstance

The **proc_CheckMeetingInstance** stored procedure is called to determine whether a given meeting instance exists in the meeting workspace 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
);
```

@WebId: The identifier of an instance of a meeting workspace. *@WebId* MUST NOT be NULL.

@ReturnSeriesItem: Specifies whether or not to return the meeting series item as a result set. 0 means that no result set will be returned.

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

@IsInstantiated: Specifies if the meeting is valid and instantiated. 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.

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

Value	Description
0	Successful completion. For an instance of a recurring meeting that has not been instantiated, output parameters *@IsInstantiated* and *@DTStartUTC* will be NULL
2	Three possible failure conditions: ■ Meeting instance found, but not instantiated. ■ Meeting Instance is a valid instance of the recurring meeting but not instantiated and

Value	Description
	ReturnSeriesItem = 0. ▪ No valid meeting event data found.
13	Unable to retrieve the meeting list. Output parameters *@IsInstantiated* and *@DTStartUTC* will be NULL.

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

▪ The *ReturnSeriesItem* parameter is set to 1.

▪ The meeting instance is a recurring event.

▪ The meeting instance is not already instantiated.

3.1.4.3.1 Meeting Event Result Set

proc_CheckMeetingInstance returns a result set which MUST contain one item which 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, see [MS-WSSFO].

bit3: Specifies whether this is a standalone or recurring meeting. 0 = standalone meeting, 1 = recurring meeting. Because this stored procedure is for a recurring meeting, this **field** MUST be 1 for the **row** returned.

tp_InstanceID: The identifier for this meeting instance. For a recurring meeting instance this is the beginning date of the meeting in UTC. For the default instance of a recurring meeting this field will be 0. Because the result set returned is always for the default instance associated with the specified meeting workspace, this MUST return a 0.

int1: Specifies the event type for this meeting as defined by Meetings Event Type (Section 2.2.2.3). By definition this field will be 1 in the row returned, indicating a recurring 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. 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 is NULL. Because the result set returned by this stored procedure is

always for the default instance associated with specified meeting workspace, this field MUST return a NULL.

datetime1: The beginning date for this meeting instance.

datetime2: The end date for this meeting instance. For the default instance which is returned by this stored procedure, this field 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 is NULL it indicates that the series has not been suppressed.

int3: The enumeration value representing the time zone used when creating this meeting. See [MS-WSSFO], section Time Zone Identifier for more detail.

3.1.4.4   proc_CheckNavStructContainsPage

The **proc_CheckNavStructContainsPage** stored procedure is called to determine if a document in a site exists in the navigation structure of the site. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_CheckNavStructContainsPage(
    @SiteId            uniqueidentifier,
    @DirName           nvarchar(256),
    @LeafName          nvarchar(128)
);
```

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

@DirName: The store-relative form directory name of the requested document. If no slash is in the store-relative form URL, the value MUST be an empty string.

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

Return Values: The stored procedure returns an integer return code which 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.

3.1.4.5   proc_DeleteAllCommentsOfAllDocs

The **proc_DeleteAllCommentsOfAllDocs stored** procedure is called to delete all Web discussion comments from a folder or site. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_DeleteAllCommentsOfAllDocs(
    @SiteId           uniqueidentifier,
    @WebId            uniqueidentifier,
    @DocDir           nvarchar(256)
);
```

@SiteId: The identifier of the site collection containing the comments to be **deleted**.

@WebId: The identifier of the site containing the comments to be deleted.

@DocDir: The store-relative URL to the folder whose Web discussion comments are to be deleted. If the parameter is NULL, all Web discussion comments in the site MUST be deleted.

Return Values: proc_DeleteAllCommentsOfAllDocs returns an integer return code which MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.4.6   proc_DeleteDocComment

The **proc_DeleteDocComment stored procedure** is called to delete a Web discussion comment from a document. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_DeleteDocComment(
    @SiteId           uniqueidentifier,
    @DocFullUrl       nvarchar(260),
    @Id               int,
    @UserTitle        nvarchar(255)
);
```

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

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

@Id: The identifier of the Web discussion comment to be deleted.

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

Return Values: proc_DeleteDocComment returns an integer return code which MUST be listed in the following table:

Value	Description
0	Successful completion.
3	Either the document does not exist in the specified site collection, or the Web discussion comment was not found with the specified document.

Result Sets: MUST NOT return any result sets.

3.1.4.7   proc_DeleteSubscription

The **proc_DeleteSubscription** stored procedure is called to delete an alert. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_DeleteSubscription(
    @SiteId          uniqueidentifier,
    @SubId           uniqueidentifier,
    @UserId          int
);
```

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

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

@UserId: The identifier as described in UserInfo table of the user for whom the alert has been created.

Return Values: proc_ DeleteSubscription returns an integer return code which MUST be listed in the following table:

Value	Description
0	Successful completion.
31	General Error

Result Sets: MUST NOT return any result sets.

3.1.4.8   proc_DeleteSubscriptionJunctionEntries

The **proc_ DeleteSubscriptionJunctionEntries** stored procedure is called to clean up junction entries for up to 256 subscription identifiers after the relevant alerts, 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
```

```
    );
```

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

Return Values: The stored procedure MUST return an integer return code that MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.4.9   proc_EnumDoclibsFileDlg

The **proc_EnumDoclibsFileDlg** stored procedure is called to enumerate certain properties of document library items in a given site.

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

```
PROCEDURE proc_EnumDoclibsFileDlg(
    @WebId                  uniqueidentifier,
    @Collation              nvarchar(32),
    @GetTemplate            bit,
    @OnlyLibsWithTemplates  bit,
    @IncludeSubWebs         bit,
    @Scopes                 image = NULL
);
```

@WebId: The identifier of the site containing the requested document libraries.

@Collation: A Windows collation name string identifier which 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 'Title' column using this collation.

@GetTemplate: A bit flag (0 or 1) 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 (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 (0 or 1) that indicates whether the result set should return properties of document libraries or properties of subsites within the site specified by *@WebId*. A value of 1 indicates that only the properties of subsites are included in the result set. A value of 0 indicates only properties of document libraries are included in the result set.

@Scopes: A binary image 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 MAY be any number of 16-byte fragments in the image. The results in the result set are restricted to document libraries matching these GUID values.

Return Values: The stored procedure MUST return an integer error code that MUST be 0.

Result Sets: MUST return one of the result sets as follows.

- If *@GetTemplate* is set to 1, then the EnumerationWithTemplates result set MUST be returned.

- If *@GetTemplate* is set to 0, then the EnumerationWithoutTemplates result set MUST be returned.

3.1.4.9.1 EnumerationWithTemplates Result Set

EnumerationWithTemplates contains certain properties of document libraries in a given site.

This result set MUST contain one or more rows if there are matches for the specified parameters, and MUST contain 0 rows 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 set to 1, this value MUST be NULL.

tp_Title: The title of the document library.

tp_Description: The description of the document library.

tp_ImageUrl: The store-relative URL of an image used to represent the document library.

3.1.4.9.2 EnumerationWithoutTemplates Result Set

EnumerationWithoutTemplates contains certain properties of document libraries in a given site**.**

This result set MUST contain one or more rows if there are matches for the specified parameters, and MUST contain 0 rows 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.

tp_Description: The description of the document library.

tp_ImageUrl: The store-relative URL of an image used to represent the document library.

3.1.4.10   proc_EnumEmailAliases

The **proc_EnumEmailAliases** stored procedure is invoked to retrieve the collection of all e-mail aliases. The T-SQL syntax for the stored procedure is as follows:

PROCEDURE proc_EnumEmailAliases();

Return Values: The stored procedure returns an integer return code which MUST be 0.

Result Sets: The stored procedure MUST return a single EmailAliases result set as defined in the EmailAliases Result Set (Section 2.2.6.1).

3.1.4.11   proc_EnumEmailAliasesBySite

The **proc_EnumEmailAliasesBySite** stored procedure is called to retrieve the collection of all 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: The stored procedure returns an integer return code which MUST be 0.

Result Sets: The stored procedure MUST return a single EmailAliases result set as defined in the EmailAliases Result Set (Section 2.2.6.1).

3.1.4.12   proc_EnumSubscribedSites

The **proc_EnumSubscribedSites** stored procedure is called to enumerate the sites that have been subscribed for alerts. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_EnumSubscribedSites(
    @NotifyFreq          int,
    @bAlwaysNotify       bit
);
```

@NotifyFreq: A Notification Frequency Type (Section 2.2.2.5) that specifies an integer indicating the frequency of the alert notification.

@bAlwaysNotify: A bit that indicates that the alert is always a notify alert.

Return Values: The stored procedure MUST return an integer return code that MUST be 0.

Result Sets: The stored procedure MUST return one result set as follows:

3.1.4.12.1   SubscribedSites Result Set

The SubscribedSites result set returns list of site identifiers that have been subscribed. The SubscribedSites result set will contain zero or more rows. The T-SQL syntax for the result set is as follows:

```
SiteId              uniqueidentifier;
```

SiteId: The unique identifier for the site.

3.1.4.13 proc_GetAlertsSqmData

The **proc_GetAlertsSqmData** stored procedure is called to retrieve the number of alerts. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetAlertsSqmData();
```

Return Values: proc_GetAlertsSqmData 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 result set.

3.1.4.13.1 AlertsSqmData Result Set

AlertsSqmData contains the number of alerts that exist. The AlertsSqmData result set MUST contain one row of five columns. The AlertsSqmData result set is defined using T-SQL syntax, as follows:

```
{NumImmedAlerts}     int,
{NumSchedAlerts}     int,
{NumAlertsItem}      int,
{NumAlertsList}      int,
{NumAlertsCustom}    int;
```

{NumImmedAlerts}: The number of immediate alerts. This value MUST NOT be NULL.

{NumSchedAlerts}: The number of scheduled alerts. This value MUST NOT be NULL.

{NumAlertsItem}: The number of alerts of type SPAlertType.Item. This value MUST NOT be NULL.

{NumAlertsList}: The number of alerts of type SPAlertType.List. This value MUST NOT be NULL.

{NumAlertsCustom}: The number of alerts of type SPAlertType.Custom. This value MUST NOT be NULL.

3.1.4.14 proc_GetDefaultMtgInstance

The **proc_GetDefaultMtgInstance** stored procedure is called to return a result set of all meeting instances tied to the supplied meeting workspace. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetDefaultMtgInstance(
    @WebId              uniqueidentifier
);
```

@WebId: The identifier of an instance of a meeting workspace.

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

Value	Description
0	Successful completion. result set MUST have one or more records.
13	No meetings list was found on the workspace specified by *@WebId*. The result set does NOT have any records and MUST be empty.

Result Sets: The stored procedure MUST return a single result set on successful completion.

3.1.4.14.1 Default Meeting Instance Result Set

The default meeting instance 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, see [MS-WSSFO].

bit3: Specifies whether this is a standalone or recurring meeting. 0 = standalone meeting, 1 = recurring meeting.

tp_InstanceID: Specifies the identifier for this meeting instance in the meeting workspace. For a recurring meeting instance this is the beginning date of the meeting in UTC. For the default instance of a recurring meeting this field will be 0. For standalone meetings, this field is 1 for the first instance and is incremented by 1 for every additional standalone meeting associated with the same meeting workspace.

int1: Specifies the event type for this meeting as defined by Meetings Event Type (Section 2.2.2.3).

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. For a meeting instance exception, this is the original starting date and time of the meeting instance. For the default meeting instance or standalone meetings, this field MUST be NULL.

datetime1: Specifies the beginning date for this meeting instance.

datetime2: The end datetime for this meeting instance. For the default instance of a recurring meeting tied to a meeting workspace, this is set to the maximum sql datetime.

int2: Specifies the duration of a meeting instance, in seconds. For recurring meetings tied to workspaces, only the default instance has the duration field set to the duration of an instance in the

series. This field is NULL for all other instances in the series. For standalone meetings tied to workspaces, this field is set to 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 MUST be NULL.

ntext2: This returns any recurrence rule specified for the recurring meeting. When specified, this field is set for the default 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 it has been suppressed until. When this field is NULL it indicates that the series has not been suppressed.

int3: The enumeration value representing the time zone used when creating this meeting. See [MS-WSSFO], Time Zone Identifier, for more detail.

3.1.4.15   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)
);
```

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

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

Return Values: An integer which MUST be 0.

Result Sets: MUST return one GetDocComments result set.

3.1.4.15.1   GetDocComments Result Set

Returns the set of Web discussion comments associated with this document. This result set contains one row 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 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 flag value that indicates the status of a Web discussion comment.

Comment: The body text of the Web discussion comment.

3.1.4.16   proc_GetEventDataAndSubscriptionFilters

The **proc_GetEventDataAndSubscriptionFilters** stored procedure is called to return events and subscriptions. The T-SQL syntax, as follows:

```
PROCEDURE proc_GetEventDataAndSubscriptionFilters();
```

Return Values: The stored procedure MUST return an integer error code that MUST be 0.

Result Sets: proc_GetEventDataAndSubscriptionFilters MUST return 3 result sets as follows.

3.1.4.16.1   Events Result Set

The Events result set MUST be returned. The T-SQL syntax for the result set is 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 identifier of the list to which the event is related to.

Id: The identifier of the event.

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

EventType: An Alert Event Type (Section 2.2.3.1) that specifies an integer mask for the types of events on which to fire the notification. This value MUST NOT be NULL.

ItemId: The identifier of the list item from the list specified by **ListId** that is associated with the event. 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 which is associated with the event, or the URL to the site, or NULL if the application already specifies the list item using *@ItemId*.

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

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

Acl: A byte array in the **access control list (ACL)** format. If this parameter is NULL, the **ACL** will be inferred from the security scope specified by the *@ScopeId* parameter. If the ScopeId parameter is also NULL, the ACL will be inferred from the *@ItemFullUrl*.

SiteId: The GUID for the site collection under which the event has occurred. This value MUST NOT be NULL.

3.1.4.16.2 Subscriptions Result Set

Subscriptions returns the subscriptions for the site and list of the events 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(64),
WebUrl              nvarchar(256),
Properties          ntext;
```

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

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

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

ItemId: The **item identifier** of the item with which this subscription is associated. This MUST NOT be NULL if the Alert Type (Section 2.2.2.2) of the subscription is item and MUST be NULL if the Alert Type (Section 2.2.2.2) of the subscription is list.

AlertType: An integer whose lower eight bits specifies the type of the alert as specified in Alert Type (Section 2.2.2.2). This MUST also contain the Special Alert Flags (Section 2.2.3.2) for an always notify alert or a system alert.

EventType: An Alert Event Type (Section 2.2.3.1) that specifies an integer mask for the types of events on which to fire the notification. This value MUST NOT be NULL.

{NotifyFreq}: A Notification Frequency Type (Section 2.2.2.5) that specifies an integer indicating the frequency of the alert notification. This value MUST NOT be NULL.

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

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

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

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

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

WebUrl: The URL of the site with which this subscription is associated. The value MUST NOT be NULL.

Properties: The properties associated with this subscription.

3.1.4.16.3 UTCTime Result Set

UTCTime returns the current time in UTC. The UTCTime result set will always return and MUST have one row. The T-SQL syntax is as follows:

```
{CurrentDateTime}    datetime;
```

{CurrentDateTime}: Contains current date and time in UTC.

3.1.4.17 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          int = 0,
    @RecurrenceIDColName    nvarchar(64),
    @RecurrenceIDRowOrdinal int = 0,
    @StartDateColName       nvarchar(64),
    @StartDateRowOrdinal    int = 0,
    @EndDateColName         nvarchar(64),
    @EndDateRowOrdinal      int = 0
);
```

@ListId: A GUID of a recurrence list (1).

@ViewStart: A datetime field corresponding to the beginning date of the recurring meeting.

@ViewEnd: A datetime field corresponding to the End Date of the recurring meeting.

@EventTypeColName: The name of the column in the UserData View that corresponds to the event type. For more information about UserData View, see [MS-WSSFO].

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

@UIDColName: The name of the column in the UserData View that corresponds to the unique identifier of the recurring meeting series.

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

@RecurrenceIDColName: The name of the column in the UserData View that corresponds to the recurrence identifier of the recurring meeting series.

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

@StartDateColName: The name of the column in the UserData View that corresponds to the beginning date of the recurring meeting series. For more information about UserData, see [MS-WSSFO].

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

@EndDateColName: The name of the column in the UserData View that corresponds to the End Date of the recurring meeting series.

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

Return Values: An integer which MUST be 0.

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

3.1.4.17.1   Exceptions Result Set

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 if the specified input parameters map to valid meeting series exceptions in the UserData View. For more information about UserData View, see [MS-WSSFO].

The Exceptions result set is empty if the column 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 is specified the **@ UIDColName** parameter. The identifier of the recurring meeting.

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

3.1.4.18   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    int = 0,
    @UIDColName             nvarchar(64),
    @UIDRowOrdinal          int = 0,
    @RecurrenceIDColName    nvarchar(64),
    @RecurrenceIDRowOrdinal int = 0,
    @StartDateColName       nvarchar(64),
    @StartDateRowOrdinal    int = 0
);
```

@ListId: A GUID of a recurrence list (1).

@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 in the UserData View that corresponds to the event type. For more information about UserData, see [MS-WSSFO].

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

@UIDColName: The name of the column in the UserData View that corresponds to the unique identifier of the recurring meeting series.

@UIDRowOrdinal: The name of the row ordinal in the UserData View that MUST contain the unique identifier of the recurrence instance in the column specified by the *@UIDColName* parameter.

@RecurrenceIDColName: The name of the column in the UserData View that corresponds to the recurrence identifier of the recurring meeting series.

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

@StartDateColName: The name of the column in the UserData View that corresponds to the beginning date of the recurring meeting series.

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

Return Values: An integer which MUST be 0.

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

3.1.4.18.1 Exceptions Result Set

Exceptions returns the list item identifier for a meeting instance and the recurrence identifier that identifies the specific meeting instance. The Exceptions result set contains one or more rows if the specified input parameters map to valid meeting series exceptions in the UserData View. The Exceptions result set is empty if the column names and ordinals specified as input parameters, exist in the UserData View but do not map to any valid meeting series exceptions. (For more information about UserData View, see [MS-WSSFO].) 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 is specified the **@RecurrenceIDColName** parameter.

3.1.4.19 proc_GetItemCountPerInstance

The **proc_GetItemCountPerInstance** stored procedure is called to retrieve the number of items within a given list associated with the given instance identifier. Items in the given list with a NULL instance identifier 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
);
```

@ListId: Specifies the identifier of the list 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 with the specified instance identifieror NULL. This value is filled and returned to the caller.

Return Values: An integer which MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.4.20 proc_GetListAndChildrenNSInfo

The **proc_GetListAndChildrenNSInfo** stored procedure is called to retrieve subfolders of a **root folder** for a list, or subfolders of a folder. The T-SQL syntax for the stored procedure is as follows:

Release: Sunday, December 19, 2010

```
PROCEDURE proc_GetListAndChildrenNSInfo(
    @SiteId          uniqueidentifier,
    @WebId           uniqueidentifier,
    @ListId          uniqueidentifier,
    @FolderId        uniqueidentifier,
    @GetList         bit = 0,
    @GetSubFolders   bit = 0
);
```

@SiteId: Specifies the identifier of the site collection in which the list exists.

@WebId: Specifies the identifer of the site in which the list exists.

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

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

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

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

Return Values: proc_GetListAndChildrenNSInfo returns an integer return code which MUST be 0

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

Returned Result Set	Input Parameter
1 ListRootFolderNSInfo result set	*@GetList*=1
1 ChildFoldersNSInfo result set	*@GetSubFolders*=1

3.1.4.20.1 ListRootFolderNSInfo Result Set

The ListRootFolderNSInfo result set contains a row set containing the information about the list, 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 if the list can be found. It MUST contain no rows if the specified list 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,
Acl                  image,
AnonymousPermMask    bigint;
```

tp_Title: The title of the list whose subfolders will be returned.

tp_Id: The list identifier of the list.

FullUrl: The complete store-relative URL for the root folder for the list.

tp_RootFolder: The identifier of the root folder for the list.

tp_ImageUrl: The URL of the image used to represent the list.

tp_BaseType: The base type of the list as defined in [MS-WSSFO], section List Base Type.

tp_ServerTemplate: The List Server Template enumeration value of the list template that defines the base structure of the list. See [MS-WSSFO], section List Server Template for more detail.

FolderChildCount: The number of subfolders of the root folder.

Acl: The binary serialization of the ACL for the root folder. The format is specified in [MS-WSSFO] section 2.2.3.3.

AnonymousPermMask: The permissions mask that applies to an anonymous user of the root folder. The format is specified in [MS-WSSFO] section 2.2.2.27.

3.1.4.20.2 ChildFoldersNSInfoResult Set

ChildFoldersNSInfo returns a rowset containing the subfolders of the root folder for the specified list if *@GetList* is set to 1 or the folder specified by *@FolderId* if *@GetList* is set to 0. The ChildFoldersNSInfo result set MUST be returned when the input parameter *@GetSubFolders* is set to 1. The ChildFoldersNSInfo result set MUST contain 0 or more rows. 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 form URL for the root folder for the subfolder.

FolderChildCount: The number of subfolders of the subfolder.

ListId: The list identifier of the list containing the folder.

Acl: The binary serialization of the ACL for the folder. The format is specified in [MS-WSSFO] section 2.2.3.3.

AnonymousPermMask: The permissions mask that applies to an anonymous user of the subfolder. The format is specified in [MS-WSSFO] section 2.2.2.27.

Release: Sunday, December 19, 2010

3.1.4.21   proc_GetMatchingSubscriptionsData

The **proc_GetMatchingSubscriptionsData** stored procedure is called to return 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
);
```

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

Return Values: MUST be 0.

Result Sets: MUST return a single result set as follows:

3.1.4.21.1 MatchingSubscriptionsData Result Set

MatchingSubscriptionsData contains the 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(64),
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);
```

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

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

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

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

WebUrl: The URL of the site with which this subscription is associated. The value MUST NOT be NULL.

ListUrl: The URL of the list with which this subscription is associated. The value MUST NOT be NULL.

WebTitle: The title of the site with which this subscription is associated.

WebLanguage: The language of the site with which this subscription is associated. The value MUST NOT be NULL.

WebLocale: The locale of the site with which this subscription is associated. The value MUST NOT be NULL.

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

WebTime24: The time format which specifies whether the time displayed for the site with which this 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 with which this 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 with which this subscription is associated. For more information, see Abstract Data Model (Section 3.2.1).The value MUST NOT be NULL.

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

ListTitle: The title of the list with which this subscription is associated. The value MUST NOT be NULL.

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

ListServerTemplate: The server template of the list with which this subscription is associated. The value MUST NOT be NULL.

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

ItemId: The item identifier with which this subscription is associated. This MUST NOT be NULL if the Alert Type (Section 2.2.2.2) of the subscription is item and MUST be NULL if the Alert Type (Section 2.2.2.2) of the subscription is list.

EventType: An Alert Event Type (Section 2.2.3.1) that specifies an integer mask for the types of events on which to fire the notification. This value MUST NOT be NULL.

Binaryfilter: Contains compiled binary syntax query in Collaborative Application Markup Language (CAML) for a filter to apply to the alert associated with this subscription.

Properties: The properties associated with this subscription.

AlertTitle: The title of the alert of the subscription.

AlertType: An integer whose lower eight bits specifies the type of the alert as specified in Alert Type (Section 2.2.2.2). This MUST also contain the Special Alert Flags (Section 2.2.3.2) for an always notify alert or a system alert.

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

3.1.4.22   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.

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
);
```

@WebId: Specifies the identifier of the site 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.

Release: Sunday, December 19, 2010

@DTEndUTC: The ending datetime the 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** of the meeting instance. This property defines the revision sequence number of the meeting instance within a sequence of revisions. This parameter is returned to the caller.

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

Value	Description
0	Successful completion.
2	No meeting instances found for the given *@WebId* and *@InstanceID*.
13	Two possible failure conditions: ▪ No meetings list found on the meeting workspace specified by *@WebId*. ▪ *@OrganizerID* or *@UID* for the meeting instance is set to NULL

Result Sets: MUST NOT return any result sets.

3.1.4.23   proc_GetRecurrenceSeriesData

The **proc_GetRecurrenceSeriesData** stored procedure is called to get Series UID, beginning date, and Master Series identifier for the given meeting series list, meeting series item, and meeting 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         int = 0,
    @UIDColName              nvarchar(64),
    @UIDRowOrd               int = 0,
    @StartDateColName        nvarchar(64),
    @StartDateRowOrd         int = 0,
    @MasterIDColName         nvarchar(64),
    @MasterIDRowOrd          int = 0,
    @SeriesUID               uniqueidentifier OUTPUT,
    @StartDate               datetime OUTPUT,
    @MasterSeriesID          int OUTPUT
);
```

@ListId: Specifies the identifier of a recurrence list.

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

@EventType: Specifies the event type to retrieve as defined by Meetings Event Type (Section 2.2.2.3)**.**

@EventTypeColName: Specifies the name of the column in the UserData View of the content database that corresponds to the event type. For more information about UserData View, see [MS-WSSFO].

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

@UIDColName: The name of the column in the UserData View of the content database that corresponds to the unique identifier of the Instance.

@UIDRowOrd: The name of the row ordinal in the UserData View of the content database that MUST contain the unique identifier of the Instance in the column specified by the *@UIDColName* parameter.

@StartDateColName: The name of the column in the UserData View of the content database that corresponds to the beginning date of the recurring instance.

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

@MasterIDColName: The name of the column in the UserData View of the content database that corresponds to the master item identifier of the recurring instance.

@MasterIDRowOrd: The ordinal of list item that MUST contain the master item identifier specified inside the column 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 MAY be NULL if the specified meeting series item is not associated with a master item identifier in the UserData View.

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

Value	Description
0	Successful completion. *@SeriesUID* and *@StartDate* MUST NOT be NULL.
4005	No records found that match the input parameters. All output parameters will be NULL.

Result Sets: MUST NOT return any result sets.

3.1.4.24   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),
```

Release: Sunday, December 19, 2010

```
    @LeafName           nvarchar(128)
);
```

@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.

Return Values: The client MUST ignore the return code returned by the stored procedure.

Result Sets: The stored procedure MUST return one result set as follows:

3.1.4.24.1   VersionMetaInfo Result Set

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

```
UnVersionedMetaInfo        image,
UnVersionedMetaInfoVersion  int;
```

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

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

3.1.4.25   proc_GetWebAndChildrenNSInfo

The **proc_GetWebAndChildrenNSInfo** stored procedure is called to retrieve a set of properties related to a site, its subsites, its document libraries and its lists. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetWebAndChildrenNSInfo(
    @SiteId            uniqueidentifier,
    @WebId             uniqueidentifier,
    @GetThisWeb        bit,
    @GetSubwebs        bit,
    @GetDocLibs        bit,
    @GetLists          bit
);
```

@SiteId: The identifier of the site collection containing the site to retrieve.

@WebId: The identifier of the site to retrieve.

@GetThisWeb: If this parameter is 1, a SiteNSInfo result set of the site specified by *@WebId* is returned. The default value is 0

@GetSubwebs: If this parameter is 1, a SiteNSInfo result set of the subsites under the site specified by *@WebId* is returned. The default value is 0.

@GetDocLibs: If this parameter is 1, a ListNSInfo result set of document library of the site specified by *@WebId* is returned. If *@GetLists* is set to 1, the rows in this result set are merged with the rows in the result set of lists. The default value is 0.

@GetLists: If this parameter is 1, a ListNSInfo result set that contains the lists except document libraries under the site specified by *@SiteId* and *@WebId* is returned. If *@GetDocLibs* is set to 1, the rows in this result set are merged with the rows in the result set of document library. The default value is 0.

Return Values: proc_GetWebAndChildrenNSInfo returns an integer return code which MUST be 0:

Result Sets: proc_GetWebAndChildrenNSInfo MUST return 0 to 3 result sets based on the values of the four bit flag input parameters: *@GetThisWeb*, *@GetSubWeb*, *@GetDocLibs* and *@GetLists*.

Returned Result Set	Input Parameter
1 SiteNSInfo result set	*@GetThisWeb*=1
1 SiteNSInfo result set	*@GetSubWeb*=1
1 ListNSInfo result set	*@GetDocLibs*=1 OR *@GetLists*=1

3.1.4.25.1   SiteNSInfo Result Set

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

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

Title: The title of the site.

Id: The identifier of the site.

FullUrl: The store-relative URL of the site.

WebTemplate: The identifier for the template used in the **Site definition** to define the base structure of this subsite. The values are the same as defined by ProvisionConfig. See [MS-WSSFO], section Site Metadata Result Set for additional details.

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

Value	Description
-1	This site has not had any template provisioned.
0	This site has the implementation-specific default template applied.
1	This site has the Team Collaboration site template applied.
2	This site has the meeting workspace site template applied.
3	This site has the **Central Administration site** template applied.
4	This site has the Wiki site template applied.
9	This site has the Blog site template applied.

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

Acl: The binary serialization of the ACL for the site. The format is specified in [MS-WSSFO] section 2.2.3.3.

AnonymousPermMask: A permission mask indicating the rights granted to a user that is anonymous, or has no specific rights, to this site. The format is specified in [MS-WSSFO] section 2.2.2.27.

3.1.4.25.2 ListNSInfo Result Set

ListNSInfo result set returns a set of properties related to all lists if *@GetLists*=1 or document libraries if *@GetDocLibs* =1 under the specified site. The ListNSInfo result set MUST contain 0 to many rows. The T-SQL syntax for the stored procedure 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,
Acl                  image,
AnonymousPermMask    bigint;
```

tp_Title: The title of the list or document library

tp_Id: The list identifier of the list or document library.

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

tp_RootFolder: The identifier of the root folder for the list or document library

tp_ImageUrl: The URL of the image used to represent the list or document library

tp_BaseType: This specifies the base type of the list or document library as defined in [MS-WSSFO], section List Base Type.

tp_ServerTemplate: The List Server Template enumeration value of the list template that defines the base structure of the list or document library. See [MS-WSSFO], section List Server Template, for more detail.

FolderChildCount: The number of subfolders of the root folder of the list or document library.

Acl: The binary serialization of the ACL for the list or document library. The format is specified in [MS-WSSFO] section 2.2.3.3.

AnonymousPermMask: The permissions mask that applies to an anonymous user of the list or document library. The format is specified in [MS-WSSFO] section 2.2.2.27.

3.1.4.26   proc_GetWebComments

The **proc_GetWebComments stored procedure** is called to retrieve all Web discussion comments for a site. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetWebComments(
    @SiteId          uniqueidentifier,
    @WebId           uniqueidentifier
);
```

@SiteId: The identifier of the site collection containing the specified site.

@WebId: The identifier of the site containing the requested Web discussion comments.

Return Values: The stored procedure returns an integer return code which MUST be in the following table:

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

Result Sets: The stored procedure MUST return the GetWebComments result set.

3.1.4.26.1   GetWebComments Result Set

Returns all Web discussion comments for the specified site. This result set MUST be returned, and contains one row 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 0.

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 flag value that indicates the status of a Web discussion comment.

Comment: The body text of the Web discussion comment.

3.1.4.27   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
);
```

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

@WebId: The site identifier of the site for which ACLs are requested.

@NavParentWebId: The site identifier of the navigational parent site of the site specified by *@WebId* This field MAY be NULL. If it is not NULL, additional rows MUST be included in the Nav Acls result set.

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

Return Values: The stored procedure returns an integer return code which MUST be ignored.

Result Sets: The stored procedure MUST return 1 result set as defined in the Nav ACLs Result Set (Section 2.2.6.2).

3.1.4.28   proc_GetWebNavStruct

The **proc_GetWebNavStruct** stored procedure is called to return navigation structure for a given site. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetWebNavStruct(
    @SiteId             uniqueidentifier,
    @WebId              uniqueidentifier,
    @Inherited          bit
);
```

@SiteId: The site collection identifier for the site collection which contains the site specified by *@WebId*.

@WebId: The identifier of the site whose navigation structure has been requested.

@Inherited: A bit specifying whether the navigation structure from the site is inherited by its subsites. If this parameter is set to 1, the site scope and CachedDataVersion result set MUST be returned.

Return Values: The stored procedure returns an integer return code which MUST be ignored.

Result Sets: The stored procedure MUST return two or three result sets as follows.

3.1.4.28.1   Site Scope and CachedDataVersion Result Set

Returns the security scope and CachedDataversion of the site. If *@Inherited* is equal to 1, then this result set MUST be returned, otherwise this result set MUST NOT be returned. If the result set returns, it MUST return one row. 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.

3.1.4.28.2   Nav ACLs Result Set

Returns the ACLs of the documents represented by navigation nodes in the navigation structure of the site. This result set MUST return zero or more rows as defined in the Nav ACLs Result Set (Section 2.2.6.2), where the *@NavParentWebId* is NULL and the *@Inherited* value is the value of the *@Inherited* parameter.

3.1.4.28.3   Nav Data Result Set

Returns the information about the navigation structure of the site. It MUST return one or more rows. The T-SQL syntax for the result set is as follows:

```
Eid                 int,
EidParent           int,
ElementType         tinyint,
{Url}               nvarchar(260),
Name                nvarchar(256),
```

Release: Sunday, December 19, 2010

```
NodeMetainfo          image,
NonNavPage            bit,
NavSequence           bit,
ChildOfSequence       bit,
DateLastModified      datetime,
{ScopeId}             uniqueidentifier,
{SecurityType}        int;
```

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

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

ElementType: This represents the Navigation Node Type (Section 2.2.2.4) 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 0.

Name: The display name of the navigation node.

NodeMetainfo: A binary serialization of the navigation node metadata.

NonNavPage: A bit representing whether the navigation node will be filtered out when rendering the navigation structure. If the bit is set to 1, then the navigation node SHOULD be filtered out<5>. If it is set to 0, it MUST NOT be filtered out.

NavSequence: A bit representing whether the navigation node represents a link bar**.** If this navigation node represents a link bar, this bit MUST be set to 1, else it MUST be set to 0.

ChildOfSequence: A bit representing whether the navigation node is a child of a navigation node which represents a link bar. If it does, this bit MUST be set to 1, else it MUST be set to 0.

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}: This defines what kind of permissions are required on the user to be able 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 that are not in a list or document library.
2	This navigation node requires the user to have permissions to view Web pages in the site that are form or **list view pages** of a list or document library.
3	This navigation node requires the user to have permissions to view Web pages in the site that are items in a document library.

3.1.4.29   proc_GetWebNavStructNodeByIds

The **proc_GetWebNavStructNodeByIds** stored procedure is called to fetch information about a single navigation node in a site. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetWebNavStructNodeByIds(
    @SiteId            uniqueidentifier,
    @WebId             uniqueidentifier,
    @Eid               int
);
```

@SiteId: The site collection identifier for the site collection which contains the site specified by *@WebId*.

@WebId: The identifier of the site which contains the navigation node.

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

Return Values: The stored procedure returns an integer return code which MUST be 0.

Result Sets: The stored procedure MUST return two result sets as follows: Nav Data Result Set (Section 2.2.6.3) and Nav Children Result Set (Section 2.2.6.4), respectively.

3.1.4.30   proc_GetWebSubscriptions

The **proc_GetWebSubscriptions** stored procedure is called to return subscriptions for a given site collection, site and user identifier. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetWebSubscriptions(
    @SiteId            uniqueidentifier,
    @WebId             uniqueidentifier,
    @UserId            int
);
```

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

@WebId: The identifier of the site for which the subscriptions are requested to be returned.

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

Return Values: MUST be 0.

Result Sets: MUST return a single result set as follows:

3.1.4.30.1   WebSubscriptions Result Set

WebSubscriptions contains the subscriptions for the site collection specified by *@SiteId*, site specified by *@WebId* and user identifier specified by *@UserId*. If *@UserId* is NULL, subscriptions are not filtered by the UserId and the result set contains all subscriptions for all users. The WebSubscriptions result set will always return and MUST contain no rows if no matching subscriptions are found; otherwise, the WebSubscriptions result set MUST contain the number of rows equal to the number of matching subscriptions. 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;
```

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

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

ItemId: The item identifier with which this subscription is associated. This MUST NOT be NULL if the Alert Type (Section 2.2.2.2) of the subscription is item and MUST be NULL if the Alert Type (Section 2.2.2.2) of the subscription is list.

EventType: An Alert Event Type (Section 2.2.3.1) that specifies an integer mask for the types of events on which to fire the notification. This value MUST NOT be NULL.

NotifyFrequency: A Notification Frequency Type (Section 2.2.2.5) that specifies an integer indicating the frequency of the alert notification. This value MUST NOT be NULL.

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

Status: An Alert Status Type (Section 2.2.2.1) that specifies an integer indicating the status of the alert subscription. This value MUST NOT be NULL.

UserId: The user identifier for whom the alert will be created. This value MUST NOT be NULL.

ListUrl: The URL of the list with which this subscription is associated. The value MUST NOT be NULL.

ListTitle: The title of the list with which this subscription is associated.

ItemName: The name of the list item associated with the subscription. This value MUST be NULL if ItemId is NULL or the subscription is for an item in a links list or if the base type of the list is Issue ([MS-WSSFO]).

AlertTitle: The title of the alert of the subscription.

AlertType: An integer whose lower eight bits specifies the type of the alert as specified in Alert Type (Section 2.2.2.2). This MUST also contain the Special Alert Flags (Section 2.2.3.2) for an always notify alert or a system alert.

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

Filter: The syntax query in Collaborative Application Markup Language for a filter to apply to the alert associated with this subscription.

BinaryFilter: The compiled binary syntax query in Collaborative Application Markup Language for a filter to apply to the alert associated with this subscription.

Properties: The properties associated with this subscription.

3.1.4.31   proc_GetWebSubscriptionsForBackup

The **proc_GetWebSubscriptionsForBackup** stored procedure is called to return subscriptions associated with a site collection and a site. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetWebSubscriptionsForBackup(
    @SiteId          uniqueidentifier,
    @WebId           uniqueidentifier
);
```

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

@WebId: The identifier of the site for which the subscriptions are requested to be returned.

Return Values: MUST be 0.

Result Sets: MUST return a single result set as follows:

3.1.4.31.1   WebSubscriptionsForBackupResult Set

WebSubscriptionsForBackup contains the subscriptions, and their associated document information, for the site collection specified in the *@SiteId* parameter and the site specified in the *@WebId* parameter. The WebSubscriptionsForBackup result set will always return and MUST contain zero rows if there are no current subscriptions for the site collection specified in the *@SiteId* parameter and the site specified in the *@WebId* parameter; otherwise, if there are current subscriptions, the WebSubscriptionsForBackup result set MUST contain the number of rows equal to the number of current 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 subscription. The value MUST NOT be NULL.

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

ItemId: The item identifier of the item referenced by the subscription. This MUST NOT be NULL if the Alert Type (Section 2.2.2.2) of the subscription is item and MUST be NULL if the Alert Type (Section 2.2.2.2) of the subscription is list.

EventType: An Alert Event Type (Section 2.2.3.1) that specifies an integer mask for the types of events on which to fire the notification. This value MUST NOT be NULL.

NotifyFrequency: A Notification Frequency Type (Section 2.2.2.5) that specifies an integer indicating the frequency of the alert notification. This value MUST NOT be NULL.

UserId: The identifier of the user. The value MUST NOT be NULL.

UserEmail: The e-mail of the user. The value MUST NOT be NULL.

DirName: The directory of the document referred to by the subscription. The value MAY be NULL if the subscription does not refer to a document.

LeafName: The leaf name (file name) of the document referred to by the subscription. The value MAY be NULL if the subscription does not refer to a document.

3.1.4.32   proc_GetWebSubscriptionsUniqueUsers

The **proc_GetWebSubscriptionsUniqueUsers** stored procedure is called to return the login name of all users with active subscriptions. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetWebSubscriptionsUniqueUsers(
    @SiteId           uniqueidentifier,
    @WebId            uniqueidentifier
);
```

@SiteId: The site collection identifier of the site collection .

@WebId: The site identifier of the site.

Return Values: Must be 0.

Result Sets: MUST return a single result set as follows:

3.1.4.32.1   WebSubscriptionsUniqueUsers Result Set

WebSubscriptionsUniqueUsers contains a list of login name for all users who have nondeleted subscriptions for the site collection specified in the *@SiteId* parameter and the site specified in the *@WebId* parameter.

The WebSubscriptionsUniqueUsers result set will always return and MUST contain zero rows if there are no nondeleted subscriptions for the site collection specified in the *@SiteId* parameter and the site specified in the *@WebId* parameter: otherwise, if there are nondeleted subscriptions, the WebSubscriptionsUniqueUsers result set MUST contain the number of distinct users for whom there are 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_Login: The login name of a user with a nondeleted subscription. This value MUST NOT be empty.

3.1.4.33   proc_HandleMtgRecurPatternChange

The **proc_HandleMtgRecurPatternChange** stored procedure is called to update the events in a meeting series when the overall recurrence pattern has changed. Specifically, three updates are made to the existing meeting events:

Release: Sunday, December 19, 2010

1. Any delete exceptions in the range [*@InstanceIDStart*, *@InstanceIDEndDel*] specified are removed

2. Any modify exceptions in the range [*@InstanceIDStart*, *@InstanceIDEndSplit*] are split from the series and become standalone 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
);
```

@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 in which the meeting series exists. *@WebId* MUST NOT be NULL.

@MtgListID: Specifies the identifier of the list 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 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.

Return Values: An integer which MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.4.34 proc_HTCreateRow

The **proc_HTCreateRow 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                   image,
    @MainFile               bit,
    @maxCacheSize           int
);
```

@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 following values:

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.

Return Values: The stored procedure MUST return an integer return code which MUST be in the following table:

Value	Description
0	Successful completion.
-4	Cache entry already exists with the same *@DocSiteId*, *@DocDirName*, *@DocLeafName* and *@TransName*.

Result Sets: MUST NOT return any result sets.

3.1.4.35   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
);
```

@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 value MUST be one of the following values:

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.

Return Values: The stored procedure MUST return an integer return code of 0.

Result Sets: The stored procedure MUST return the File result set.

3.1.4.35.1 File Result Set

Returns the file that uniquely matches the input parameters. This result set MUST return a single row containing the file, if a match for the file is found. Otherwise, it MUST return zero rows. The T-SQL syntax for the result set is as follows:

```
  {File}            image;
```

{File}: The file's content.

3.1.4.36 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
  );
```

@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 following values:

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.

Return Values: An integer which MUST be 0.

Result Sets: The stored procedure MUST return Translation Name result set.

3.1.4.36.1   Translation Name Result Set

Returns the name of the main file for the set translated files in the HTML translate cache associated with the original document. This result set SHOULD contain one row if a corresponding row exists. The client MUST use the data in the first row and ignore the rest<7>. If none exist, it MUST return zero rows. 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.

3.1.4.37   proc_InstantiateMtgSeriesOccurrence

The **proc_InstantiateMtgSeriesOccurrence** 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
);
```

@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 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 beginning date of the meeting series in UTC.

@CreateOrphaned: Specifies whether the occurrence should be orphaned from the meeting series. 0 = do not orphan the occurrence (default), 1 = orphan the occurrence.

@ForAttendees: Specifies whether to copy over attendee series data as well to the new occurrence. 1 = copy only attendees, 0 = copy all except attendees, NULL = copy all (default).

@AlreadyInstantiated: Specifies the output parameter which will be set to 1 if the meeting was already instantiated. Otherwise it will be set to 0.

Return Values: An integer which 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 *@SiteId* and *@WebId* could not be created.
87	Invalid parameter encountered when creating the meeting series item. *@SiteId* or *@WebId* contains invalid data.
13	Two possible failure conditions: ▪ No meeting series found for the given *@WebId* ▪ List name not specified in default instance
80	The meeting occurrence was already instantiated.
160	Bad argument used to create the meeting series item. *@SiteId* or *@WebId* contains invalid data.
183	An instance for this meeting already exists where the 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 **read-only mode**.
1150	Unable to create the new occurrence; a newer version of Windows is required
1816	Disk quota error. The quota for site collection has reached the maximum allowable limit.

Result Sets: MUST NOT return any result sets.

3.1.4.38   proc_ListDocsHavingComments

The **proc_ListDocsHavingComments stored procedure** is called to get the collection of documents in a given folder that have 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)
);
```

@SiteId: The identifier of the site collection containing the documents whose Web discussion comments are to be retrieved.

@WebUrl: The store-relative URL to the site 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 MUST be returned.

@Collation: A **Windows collation name** string identifier which 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.

Return Values: The stored procedure returns an integer **return code** which MUST be in the following table:

Value	Description
0	Successful completion.
3	The site identified by *@SiteId* and *@WebUrl* could not be found.

Result Sets: The stored procedure MUST return the ListDocsHavingComments result set when the return code is 0.

3.1.4.38.1 ListDocsHavingComments Result Set

Returns the set of documents in the specified location containing Web discussion comments, along with the count of Web discussion comments on each document. This result set contains one row 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.

3.1.4.39 proc_MatchSchedSubscriptions

The **proc_MatchSchedSubscriptions** stored procedure is called to return scheduled subscriptions that are due and for which alert 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 OUTPUT
);
```

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

@NotifyFreq: A Notification Frequency Type (section 2.2.2.5) that specifies an integer indicating the frequency of the alert notification.

@bAlwaysNotify: A bit indicating whether the event information is returned along with the subscriptions that are requested to be returned. The value MUST be listed in the following table.

Value	Description
0	Return event information.
1	Return only subscription information.

@@EventTime: A datetime indicating the current time in UTC.

Return Values: proc_MatchSchedSubscriptions MUST return an integer that MUST be 0.

Result Sets: **proc_MatchSchedSubscriptions** MUST return a single result set as follows:

3.1.4.39.1   SchedSubscriptions Result Set

SchedSubscriptions contains the subscriptions for the site collection specified by *@SiteId* and notify frequency specified by *@NotifyFreq*. The notify frequency is the interval at which alert 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(64),
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,
```

Release: Sunday, December 19, 2010

```
{ItemId}                      int,
NotifyTime                    datetime,
NotifyFreq                    int,
Properties                    ntext,
AlertTitle                    nvarchar(1000),
AlertType                     int,
AlertTemplateName             nvarchar(255),
{EventData}                   image,
{LookupFieldPermissionResults}  image;
```

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

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

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

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

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

WebUrl: The URL of the site with which this subscription is associated. The value MUST NOT be NULL.

ListUrl: The URL of the list with which this subscription is associated. The value MUST NOT be NULL.

WebTitle: The title of the site with which this subscription is associated.

WebLanguage: The language of the site with which this subscription is associated. The value MUST NOT be NULL.

WebLocale: The locale of the site with which this subscription is associated. The value MUST NOT be NULL.

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

WebTime24: The Time format which specifies whether the time displayed for the site with which this 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 with which this 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 with which this subscription is associated. For more information, see **Abstract Data Model** (section 3.2.1). The value MUST NOT be NULL.

ListTitle: The title of the list with which this subscription is associated. The value MUST NOT be NULL.

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

ListServerTemplate: The server template of the list with which this subscription is associated. The value MUST NOT be NULL.

{ItemName}: A string that specifies the name of the object under which this event has occurred. The value MUST be NULL when *@bAlwaysNotify* is not set to 0.

{ItemFullUrl}: A string that specifies the **server-relative URL** of the object under which this event has occurred. The value MUST be NULL when *@bAlwaysNotify* is not set to 0.

{ModifiedBy}: A string which specifies the login name of a security principal who added this event. The value MUST be NULL when *@bAlwaysNotify* is not set to 0.

{TimeLastModified}: A Time Stamp in **UTC** that specifies the time when this event was last modified. The value MUST be NULL when *@bAlwaysNotify* is not set to 0.

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

{Id}: The identifier of this event in the **change log**. The value MUST be NULL when *@bAlwaysNotify* is not set to 0.

EventType: This is specified in **Alert Event Type** (section 2.2.3.1).

{EventType}: A 4-byte unsigned integer bit mask that specifies the type of an event. 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, Event Type Flags. The value MUST be NULL when *@bAlwaysNotify* is not set to 0.

{ItemId}: The integer identifier of a list item on which the event has occurred. The value MUST be NULL when *@bAlwaysNotify* is not set to 0.

NotifyTime: The time the alert is scheduled for.

NotifyFreq: A **Notification Frequency Type** (section 2.2.2.5) that specifies an integer indicating the frequency of the alert notification. This value MUST NOT be NULL.

Properties: The properties associated with this subscription.

AlertTitle: The title of the alert of the subscription.

AlertType: An integer whose lower eight bits specifies the type of the alert as specified in **Alert Type** (section 2.2.2.2). This MUST also contain the **Special Alert Flags** (section 2.2.3.2) for an always notify alert or a system alert.

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

{EventData}: Contains implementation-specific event data significant to the front-end Web server but otherwise opaque to the back-end database server. This data will be stored in the change log in the back-end database server along with this event. The value MUST be NULL when *@bAlwaysNotify* is not set to 0.

{LookupFieldPermissionResults}: Contains implementation-specific data over the permissions the user has for lookup fields. The value MUST be NULL when *@bAlwaysNotify* is not set to 0.

3.1.4.40   proc_ModifySubscription

The **proc_ModifySubscription** stored procedure is called to modify an existing 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(64),
    @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             ntext,
    @UserEmail              nvarchar(255) = NULL OUTPUT,
    @ItemName               nvarchar(255) = NULL OUTPUT,
    @ItemId                 int OUTPUT
);
```

@SiteId: The site collection identifier of the site collection in which the list or item exists**.** This value MUST NOT be NULL.

@SubId: The globally unique identifier (GUID) of the subscription that is being modified. This value MUST NOT be NULL.

@NotifyFreq: A Notification Frequency Type (Section 2.2.2.5) that specifies an integer indicating the frequency of the alert notification.

@NotifyTime: The time the alert is scheduled for. This value MUST NOT be NULL.

@NotifyTimeUTC: The time in UTC when the alert is supposed to be sent. This value MUST NOT be NULL.

@Status: An Alert Status Type (Section 2.2.2.1) that specifies an integer indicating the status of the alert subscription. This value MUST NOT be NULL.

@WebId: The globally unique identifier (GUID) of the site in which the list or item exist. This value MUST NOT be NULL.

@EventType: An Alert Event Type (Section 2.2.3.1) that specifies an integer mask for the types of events on which to fire the notification. This value MUST NOT be NULL.

@UserId: The user identifier of the user for whom the alert will be created. This value MUST NOT be NULL.

Release: Sunday, December 19, 2010

@SiteUrl: The URL of the site collection in which the list or item exists. This value MUST NOT be NULL.

@WebUrl: The URL of the site in which the list or item exists. This value MUST NOT be NULL.

@WebTitle: The title of the site in which the list or item exists.

@WebLanguage: The LCID of the display language of the site containing the list or item. This value MUST NOT be NULL.

@WebLocale: An Integer representing the LCID of the site locale. This value MUST NOT be NULL.

@WebTimeZone: An Integer representing the Time Zone of the site. 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 format for the site. This value MUST NOT be NULL.

@WebAdjustHijriDays: The number of days to extend or reduce the current month in Hijri calendars on the site with which this subscription is associated. For more information, see Abstract Data Model (Section 3.2.1).This value MUST NOT be NULL.

@AlertTitle: The title of the alert of the subscription.

@AlertType: An integer whose lower eight bits specifies the type of the alert as specified in Alert Type (Section 2.2.2.2). This MUST also contain the Special Alert Flags (Section 2.2.3.2) for an always notify alert or a system alert. This value MUST NOT be NULL.

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

@Filter: The syntax query in Collaborative Application Markup Language for a filter to apply to the alert associated with this subscription.

@BinaryFilter: The compiled binary syntax query in Collaborative Application Markup Language for a filter to apply to the alert associated with this subscription.

@Properties: The properties associated with this subscription.

@UserEmail: The e-mail address of the user.

@ItemName: The name of the item associated with the subscription.

@ItemId: The item identifier associated with the subscription. This MUST NOT be NULL if the AlertType of the subscription is Item (Section 2.2.2.2) and MUST be NULL if the AlertType of the subscription is List (Section 2.2.2.2).

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

Value	Description
0	Successful completion.
31	Error finding or modifying the subscription.

Result Sets: MUST NOT return any result sets.

3.1.4.41 proc_MoveDataToOccurrence

The **proc_MoveDataToOccurrence** stored procedure is called to move meeting instance data to the specified occurrence. This may be used to fix up orphaned instances. An occurrence can be orphaned when a series is instantiated and then the recurrence pattern is changed, making the RecurrenceID (instance identifier) 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
);
```

@UserId: An identifier for the user that is requesting the operation. This value MUST refer to an existing user identifier for the specified site.

@UserTitle: A string containing the title of the user who wishes to perform the operation.

@SiteId: The site collection identifier of the site collection containing the meeting series.

@WebId: The site identifier of the site 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 set to -3, the instance data will not be copied to any target occurrence, but will be removed from the source occurrence.

@CheckFromInstanceID: A bit parameter that specifies whether validation on the From Instance is performed or not. If set to 1, validation on the From Instance is performed.

@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**.**

@CreateOrphaned: A bit parameter that specifies whether the target instance should be orphaned from the new series.

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

Value	Description
0	Successful completion.
2	Source Instance was not found, was not orphaned, or contained invalid event type data.
3	The list item corresponding to the From Instance does not exist.
5	Access denied. The **current user** specified by *@UserId* parameter is not same as the author of the list item of the From Instance when the instance restriction level is 1.
13	Data was NOT moved to the target occurrence because source and target instances were the same AND CheckFromInstanceID, OverwriteToInstance, OR CreateMeetingEntry is 1. Also, if OverwriteToInstance AND CreateMeetingEntry are both 1 OR OverwriteToInstance is 1 when ToInstanceId is -3. Also when the target instance was not found.
33	Attempt to delete directories that contain **checked out** files in the From Instance. It is returned by proc_DropListRecord. See [MS-WSSDLIM].
87	Invalid parameter encountered when creating the meeting series item. *@SiteId* or *@WebId* contains invalid data. Returned by proc_InstantiateMtgSeriesOccurrence (Section 3.1.4.37).
1150	Concurrency violation during proc_DropListRecord. See [MS-WSSDLIM].

Result Sets: MUST NOT return any result sets.

3.1.4.42   proc_NavStructAddNewNodeByDocId

The **proc_NavStructAddNewNodeByDocId** stored procedure is called to add a new navigation node that points to an existing document in the site to 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),
    @NodeMetainfo     image,
    @NonNavPage       bit,
    @NavSequence      bit,
    @EidBase          int,
    @@EidHome         int OUTPUT
);
```

@SiteId: The site collection identifier of the site collection which contains the site.

@WebId: The site identifier of the site to which the navigation node is to be added.

@Eid: The navigation node element identifier of the new navigation node. This MUST NOT be NULL

@EidParent: The navigation node element identifier of the navigation node under which the new navigation node is being added.

[MS-WSSEUX] — v20101219
Windows SharePoint Services: Content Database End-User Experience Communications Protocol Specification

Copyright © 2010 Microsoft Corporation.

Release: Sunday, December 19, 2010

@EidPrevSib: This MUST be either the navigation node element identifier of the navigation node which MUST precede the new navigation node or a value in 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: This represents the Navigation Node Type (Section 2.2.2.4) of the navigation node. This value MUST be 0. 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

@NodeMetainfo: A binary serialization of the navigation node metadata.

@NonNavPage: A bit specifying whether the navigation node will be filtered out when rendering the navigation structure. If this is set to 1, the navigation node SHOULD be filtered out<8>. This MUST NOT be NULL.

@NavSequence: A bit specifying whether the node represents a link bar. This value MUST be 0. 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**. (specified in section 3.1.4.44.) before this stored procedure is called.

@@EidHome: If the new navigation node points to the home page of the site 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, else it MUST return NULL.

Return Values: proc_NavStructAddNewNodeByDocId returns an integer return code which MUST be listed in the following table:

Value	Description
0	Successful completion.
10001	Either *@SiteId* or *@WebId* is NULL
10002	Navigation node specified by *@EidParent* does not exist.
10006	The new navigation node was not added.
10007	The document specified by *@DocId* does not exist.
10008	The document specified by *@DocId* has no **published** version.

Result Sets: proc_NavStructAddNewNodeByDocId MUST return zero or two result sets. If the return code is 0, two result sets MUST be returned; otherwise no result set MUST be returned.

3.1.4.42.1   Nav Data Result Set

Returns information about the new added navigation node. If returned, this result set MUST return one row as defined in the Nav Data Result Set (Section 2.2.6.3).

3.1.4.42.2   Nav Children Result Set

Returns the navigation node element identifier information of all the subnodes of the new navigation node. If returned, this result set MUST contain no rows as defined in the Nav Children Result Set (Section 2.2.6.4).

3.1.4.43   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 navigation structure. 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),
    @NodeMetainfo           image,
    @IgnoreIfExists         bit,
    @NonNavPage             bit,
    @NavSequence            bit,
    @DateLastModified       datetime,
    @EidBase                int,
    @@EidHome               int OUTPUT
);
```

@SiteId: The site collection identifier of the site collection which contains the site.

@WebId: The site identifier of the site to which the navigation node is to be added.

@Eid: The navigation node element identifier of the new navigation node. This MUST NOT be NULL.

@EidParent: The navigation node element identifier of the navigation node under which the new navigation node is being added.

@EidPrevSib: This MUST be either the navigation node element identifier of the navigation node which MUST precede the new navigation node or a value in 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: This represents the Navigation Node Type (Section 2.2.2.4) of the navigation node. This MUST NOT be NULL.

@Url: The URL of the navigation node being added. If the navigation node is based on the identifier of a document in the site collection, this MUST be in store-relative form.

@Name: The display name of the navigation node. This MUST NOT be NULL.

Release: Sunday, December 19, 2010

@NodeMetainfo: A binary serialization of the navigation node metadata.

@IgnoreIfExists: A bit specifying whether to ignore adding the new navigation node if one specified by *@Eid* already exists. If this is set to 1, it MUST ensure that a navigation node with the specified *@Eid* exists and add it only if one does not exist.

@NonNavPage: A bit specifying whether the navigation node will be filtered out when rendering the navigation structure. If this is set to 1, the navigation node SHOULD be filtered out<9>. This MUST NOT be NULL.

@NavSequence: A bit specifying whether the node represents a link bar. If this is set to 1, the navigation node MUST represent a link bar. This MUST NOT be NULL.

@DateLastModified: A datetime value specifying the time when this navigation node got 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 3.1.4.44) before this stored procedure is called.

@@EidHome: If the new navigation node points to the home page of the site 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, else it MUST return NULL.

Return Values: proc_NavStructAddNewNodeByUrl returns an integer return code which MUST be listed in the following table:

Value	Description
0	Successful completion.
10001	Either *@SiteId* or *@WebId* is NULL
10002	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.

Return Values: **proc_ NavStructAddNewNodeByUrl** MUST return 0 or 2.

Result Sets: If the return code is 0, two result sets MUST be returned; otherwise no result set MUST be returned.

3.1.4.43.1   Nav Data Result Set

Returns information about the new added navigation node. If returned, this result set MUST return one row as defined in the Nav Data Result Set (Section 2.2.6.3).

3.1.4.43.2   Nav Children Result Set

Returns the navigation node element identifier information of all the subnodes of the new navigation node. If returned, this result set MUST contain no rows as defined in the Nav Children Result Set (Section 2.2.6.4).

Release: Sunday, December 19, 2010

3.1.4.44 proc_NavStructAllocateEidBlockWebId

The **proc_NavStructAllocateEidBlockWebId** stored procedure is invoked to reserve a set of navigation node element identifiers for new navigation nodes or navigation nodes that will be moved under a different navigation node. This stored procedure MUST update the site's next available navigation node element identifier. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_NavStructAllocateEidBlockWebId(
    @WebId              uniqueidentifier,
    @TempEidCount       int,
    @EidMaxNew          int,
    @@EidBase           int OUTPUT
);
```

@WebId: The site collection identifier of the site for which navigation node element identifiers are being reserved.

@TempEidCount: The number of new navigation node element identifiers that need to be reserved. This value MUST be greater than or equal to 0.

@EidMaxNew: The maximum value of the navigation node element identifier for the navigation nodes which are being added or moved along with this call.

@@EidBase: An output parameter containing the base value of navigation node element identifier to be used by the subsequent calls to update the navigation structure.

Return Values: proc_NavStructAllocateEidBlockWebId returns an integer return code which MUST be listed in the following table:

Value	Description
0	Successful completion.
10001	The site specified by *@WebId* does not exist and *@EidBase* is NULL
10004	The next available navigation node element identifier value for this site could not be updated.

Result Sets: MUST NOT return any result sets.

3.1.4.45 proc_NavStructClear

The **proc_NavStructClear** stored procedure is called to delete the navigation structure of a site and any references to these navigation nodes**.** The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_NavStructClear(
    @SiteId             uniqueidentifier,
    @WebId              uniqueidentifier
);
```

@SiteId: The site collection identifier of the site collection which contains the site.

@WebId: The site identifier of the site for which the navigation structure needs to be deleted.

Return Values: proc_NavStructClear returns an integer return code which MUST be ignored.

Result Sets: MUST NOT return any result sets.

3.1.4.46   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
);
```

@SiteId: The site collection identifier for the site collection which contains the site **specified** by *@WebId*.

@WebId: The site identifier of the site which 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 when this navigation node was changed.

@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 got deleted. If the return value is 10004, this represents the time when the navigation node was last changed.

Return Values: proc_NavStructDeleteNodeByIds returns an integer return code which 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 *@DateModified*.

Result Sets: MUST NOT return any result sets.

3.1.4.47   proc_NavStructLogChangesAndUpdateSiteChangedTime

The **proc_NavStructLogChangesAndUpdateSiteChangedTime** stored procedure is called to update the change log with the site navigation changes at the specified time. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_NavStructLogChangesAndUpdateSiteChangedTime(
    @SiteId                    uniqueidentifier,
    @WebId                     uniqueidentifier,
    @TimeLastModifiedIncoming  datetime
);
```

@SiteId: The site collection identifier of the site collection containing the change log.

@WebId: The site identifier of the site containing the change log.

@TimeLastModifiedIncoming: The date and time of the last update. If this parameter is NULL, then the current date and time MUST be set.

Return Values: The client MUST ignore the return code returned by the stored procedure.

Result Sets: MUST NOT return any result sets.

3.1.4.48   proc_NavStructMoveNode

The **proc_NavStructMoveNode** stored procedure is called to move a navigation node and all of its subnodes from under one parent navigation node to another. 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
);
```

@SiteId: The site collection identifier of the site collection which contains the site specified by *@WebId*.

@WebId: The site identifier of the site which 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 which will be the new parent of the navigation node being moved.

@EidPrevSib: The navigation node element identifier of the navigation node which MUST precede the navigation node or a value in 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.

@EidBase: The base value to use if the *@Eid* value is a temporary value. This MUST be the value returned by proc_NavStructAllocateEidBlockWebId (Section 3.1.4.44) before this stored procedure is called.

@EidHome: The navigation node element identifier of the navigation node that points to the home page of the site.

@@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 got moved. If the return code is 10004, this represents the time when the navigation node was last changed.

Return Values: proc_NavStructMoveNode returns an integer return code which MUST be listed in the following table:

Value	Description
0	Successful completion.
10004	Navigation node has been changed since the time specified by *@DateModified*.
10005	Navigation node specified by *@Eid* does not exist for the site specified by *@WebId* and *@SiteId*.
10002	Navigation node specified by *@EidParentNew* does not exist.
10006	Navigation node could not be moved.

Result Sets: proc_NavStructMoveNode MUST return zero or two result sets. If the return code is 0, then two result sets MUST be returned; otherwise no result set MUST be returned.

3.1.4.48.1 Nav Data Result Set

Returns information about the new added navigation node. If returned, this result set MUST return one row as defined in the Nav Data Result Set (Section 2.2.6.3).

3.1.4.48.2 Nav Children Result Set

Returns the navigation node element identifier information of all the subnodes of the new navigation node. If returned, this result set MUST contain no rows as defined in the Nav Children Result Set (Section 2.2.6.4).

3.1.4.49 proc_OrphanRecurringEventExceptions

The **proc_OrphanRecurringEventExceptions** stored procedure is called to orphan the meeting instances in the database that correspond to exceptions.

The T-SQL of 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              int = 0
```

Release: Sunday, December 19, 2010

```
);
```

@SiteId: The site collection identifier for the site collection containing the list that the specified item is being added to.

@ListID: The list identifier of the list that the specified item is being added to.

@WebId: The site identifier for the site containing the list that the specified item is being added to.

@UID: A GUID identifying a particular meeting series.

@ServerTemplate: The meetings list**.**

@SeriesItemID: The identifier of the meeting instance list item corresponding to the master instance of the meeting series.

@UserID: The integer identifier of the current user**.**

@RecurrenceColName: The name of the column in the UserData View that corresponds to the recurrence identifier of the recurring meeting series.

@RecurrenceRowOrd: The ordinal of the list (1) item that SHOULD contain the recurrence identifier specified inside the column identified by the parameter *@RecurrenceColName*.

@EventTypeColName: The name of the column in the UserData View that corresponds to the event type. For more information about UserData View, see [MS-WSSFO].

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

@RecDataColName: The name of the column in the UserData View that corresponds to the recurrence data of the recurring meeting series.

@ RecDataRowOrd: The ordinal of the list (1) item that SHOULD contain the recurrence data specified inside the column identified by the parameter *@RecDataColName*.

@UIDColName: The name of the column in the UserData View that corresponds to the unique identifier of the recurring meeting series.

@UIDRowOrd: The name of the row ordinal in the UserData View that contains the unique identifier of the recurring meeting series in the column specified by the *@UIDColName* parameter.

Return Values: An integer which MUST be 0<10>.

Result Sets: MUST NOT return any result set.

3.1.4.50   proc_ProcessDelMtgAttendeeListItem

The **proc_ProcessDelMtgAttendeeListItem** stored procedure is called to remove a meeting attendee list item from a meetings list. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ProcessDelMtgAttendeeListItem(
    @SiteId            uniqueidentifier,
    @WebId             uniqueidentifier,
    @ListId            uniqueidentifier,
    @ItemId            int
```

```
    );
```

@SiteId: The site collection identifier of a site collection.

@WebId: The site identifier of a site within the site collection.

@ListId: The list identifier of an attendees list in the site.

@ItemId: The identifier of the list item corresponding to the item for an attendee.

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

Value	Description
0	Successful completion.
13	Attendee was not found in list.

Result Sets: MAY return one result set as follows.

3.1.4.50.1 Site Audit Mask Result Set

The Site Audit Mask result set returns the information about the Audit Flags associated with the specified site. The Site Audit Mask result set is returned if and only if a single attendee match is found, and there exists a site defining the roles on the specified site. The Site Audit Mask result set MUST return a single row.

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

```
{Id}                  uniqueidentifier,
{AuditFlags}          int,
{InheritAuditFlags}   int,
{GlobalAuditMask}     int;
```

{Id}: The identifier of the specified site. This MUST be the same as *@WebId*.

{AuditFlags}: An Audit Flags value determining the operations to be tracked on this site. This value MUST be NULL if the site cannot be found within the site collection.

{InheritAuditFlags}: An Audit Flags value determining the operations to be tracked on one of this site's subsites. This value MUST be NULL if the site cannot be found within the site collection.

{GlobalAuditMask}: An Audit Flags value determining the operations to be tracked across the site collection that contains this site. This value MUST be NULL if the site collection cannot be found.

3.1.4.51 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           image = NULL,
        @NonNavPage             bit = NULL,
        @NavSequence            bit = NULL,
        @WantOldName            bit = 0,
        @@DateModifiedReally    datetime = OUTPUT
    );
```

@SiteId: The site collection identifier of the site collection which contains the site specified by *@WebId*.

@WebId: The site identifier of the site 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 3.1.4.44) before this stored procedure is called.

@EidHome: The navigation node element identifier of the navigation node that points to the home page of the site.

@DateModified: A datetime value specifying the last known time when this navigation node was changed.

@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 is of navigation node type 1.

@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 is of navigation node type 1.

 @UrlExternal: The URL that this navigation node represents. This MUST be ignored if the navigation node is of navigation node type 0.

@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 navigation node metadata**.** If this is set to 1, the navigation node metadata MUST be set to the value specified by **@NodeMetainfo**, else the navigation node metadata MUST not be changed.

@NodeMetainfo: A binary serialization of the navigation node metadata.

@NonNavPage: A bit specifying whether the navigation node will be filtered out when rendering the navigation structure. If this is set to 1, the navigation node SHOULD be filtered out<11>. If this is NULL, then the corresponding setting for the navigation node MUST NOT be changed.

@NavSequence: A bit representing whether the node represents a link bar. If this navigation node represents a link bar, this bit MUST be set to 1, else it MUST be set to 0. If the navigation node represented by @Eid is of Navigation Node Type 0, then 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 set to 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.

 Return Values: proc_PutWebNavStructNode returns an integer return code which MUST be listed in the following table:

Value	Description
0	Successful completion.
10005	Navigation node specified by *@Eid* does not exist for the site specified by *@WebId* and *@SiteId*.
10004	Navigation node has been changed since the time specified by *@DateModified*.
10007	The document specified by *@DirName* and *@LeafName* does not exist.
10006	The navigation node was not updated.

Result Sets: proc_PutWebNavStructNode MUST return zero, two, or three result sets. If the return code is anything other than 0, no result set MUST be returned. If the return code is 0, then either two or three result sets MUST be returned.

3.1.4.51.1 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 0, then 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. 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.

3.1.4.51.2 Nav Data Result Set

Returns information about the updated navigation node. If returned, this result set MUST return one row as defined in the Nav Data Result Set (Section 2.2.6.3).

3.1.4.51.3 Nav Children Result Set

Returns the navigation node element identifier information of all the subnodes of the new navigation node. If returned, this result set MUST contain zero or more rows as defined in the Nav Children Result Set (Section 2.2.6.4).

3.1.4.52   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
);
```

@MeetingsListId: A GUID referring to the meetings list.

@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.

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.

3.1.4.53   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               ntext,
    @Size                  int,
    @UserTitle             nvarchar(255)
);
```

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

@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 flag value that indicates the status of a Web discussion comment. If NULL, the status MUST NOT be changed.

Release: Sunday, December 19, 2010

@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.

Return Values: The stored procedure returns an integer return code which MUST be in the following table:

Value	Description
0	Successful completion.
3	The specified Web discussion comment or document does not exist.
212	The specified site collection has been locked, and writes are disallowed.
1816	The quota for the specified site collection has been exceeded.

Result Sets: MUST NOT return any result sets.

3.1.4.54   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         image,
    @MetaInfoSize     int,
    @UpdateHCD        int,
    @HCD              int,
    @UpdatedVersion   int = 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: If this parameter is NULL or equal to 0, *@HCD*, Has Copy Destination, parameter MUST be ignored.

@HCD: If this parameter is equal to 0, tp_HasCopyDestinations of the corresponding AllUserData is set to 0. Otherwise, it is set to 1.

@UpdatedVersion: The new version of the metadata.

Release: Sunday, December 19, 2010

Return Values: proc_SetVersionIndependentMetaInfo returns an integer return code which 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.

3.1.4.55   proc_SetWebNavStructInheritance

proc_SetWebNavStructInheritance is invoked to update a site's 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
);
```

@WebId: The site identifier whose inheritable navigation structure is to be updated.

@Inherits: a bit value indicating whether the site inherits navigation structure from parent site or not. When set to 1, the site specified by *@WebId* MUST inherit its inheritable navigation structure from its parent site. When set to 0, the site specified by *@WebId* MUST have a unique inheritable navigation structure.

@@NavParentWebId: Returns the GUID of new navigation parent site. When *@Inherits* is 1, SHOULD return *NavParentWebId* of the parent site, unless the *NavParentWebId* of the parent site is NULL. In that case, return the GUID of the parent site. When *@Inherits* is 0, MUST return NULL.

Return Values: proc_SetWebNavStructInheritance returns an integer return code which MUST be listed in the following table:

Value	Description
0	Successful completion.
1	The Web does not exist

Result Sets: MUST NOT return any result sets.

3.1.4.56   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,
```

Release: Sunday, December 19, 2010

```
    @InstanceID      int,
    @UserId          int
);
```

@WebId: Specifies a site identifier for an instance of a meeting workspace. The value MUST be a globally unique identifier (GUID).

@InstanceID: Specifies the identifier of the instance of the meeting that will be re-associated to the meeting series.

@UserID: Specifies the identifier of the user performing the action.

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

Value	Description
0	Successful completion.
2	Either a meeting instance with *@InstanceId* was not found, or the InstanceId is mapped to more than one row in the UserData View. For more information about UserData View, see [MS-WSSFO].
13	No meetings list was found on the workspace specified by *@WebId*.

Result Sets: MUST NOT return any result sets.

3.1.4.57   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             image,
    @CachedInheritedNavData    image,
    @CachedNavScopeData        ntext,
    @NavParentCachedNavData    image,
    @NavParentCachedScopeData  ntext,
    @CachedNavDirty            int,
    @CachedDataVersion         int,
    @NavParentCachedDataVersion   int
);
```

@WebId: The site identifier for which the navigation cache is updated**.**

@NavParentWebId: The site identifier of the navigational parent site**.**

@CachedNavData: The node structure in binary data format from the cached navigation. This binary format is specified as follows:

Navigation Node Element Identifier
NULL

Release: Sunday, December 19, 2010

Navigation Node Element Identifier
Navigation Node Element Identifier of the parent Navigation Node
NULL
Navigation Node Element of the URL of the Navigation Node
NULL
Scope Identifier
NULL
Security Type
NULL

Each of these elements is followed by a NULL inside the binary data. All the elements in the structure are represented as strings. The identifiers are converted into strings, whereas GUID values are already strings by definition. Security type is an integer value that is also converted into a string at concatenation time.

Security Type is a value that defines the **permissions** required to see this navigation node. (See [MS-WSSFO] for a list of permissions and their descriptions.) 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 that are not in a list or document library.
2	This navigation node requires the user to have permissions to view Web pages in the site that are form or list view pages of a list or document library.
3	This navigation node requires the user to have permissions to view Web pages in the site that are items in a document library.

@CachedInheritedNavData: The node structure in binary data format from the navigational parent site. For the format, please refer to the description of parameter *@CachedNavData*.

@CachedNavScopeData: The string that contains all the distinct scope identifier of all the navigation nodes of the site. Every element is included between 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. For the format, please refer to the description of parameter *@CachedNavData*.

@NavParentCachedScopeData: The string that contains all the distinct scope identifier of all the navigation nodes of the navigational parent site. The format of the string is specified in the description of parameter *@CachedNavScopeData*.

@CachedNavDirty: This MUST be 0.

@CachedDataVersion: The integer that defines the version of the cached data.

@NavParentCachedDataVersion: The integer that defines the version of the cached data of the navigational parent site.

Return Values: The stored procedure returns an integer return code which MUST be in the following table:

Value	Description
0	Successful completion.
1150	The cached data could not be updated. This MAY be caused by a concurrency violation.

Result Sets: MUST NOT return any result sets.

3.1.4.58   proc_UpdateItemJunctionsCurrentVersion

The **proc_UpdateItemJunctionsCurrentVersion** stored procedure is called to update the UIVersion of a subscribed event 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
);
```

@SiteId: Specifies the site collection identifier of the site collection in which the subscribed-to event exists.

@DirName: Specifies the directory name of the subscribed event.

@LeafName: Specifies the leaf name (file name) of the subscribed event.

@NewUIVersion: Specifies the integer to which the UI versions of item junctions specified by *@SiteId*, *@DirName*, *@LeafName* and *@Level* MUST be updated.

@Level: Specifies the status of the document for the subscribed event. The value MUST be listed in the following table:

Value	Description
255	Checked out
2	Draft.
1	Publish.
0	Unused.

Return Values: proc_UpdateItemJunctionsCurrentVersion returns an integer which MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.4.59   proc_UpdateListNavNode

The **proc_UpdateListNavNode** stored procedure is called to update the navigation node that represents a list in the navigation structure of a site. 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
);
```

@WebId: The site identifier of the site which contains the list specified by *@ListId*.

@ListId: The list identifier of the list whose navigation node is to be updated.

@ViewId: The identifier of the **default list view** of the list.

@Name: The display name of the navigation node.

@**NaVBArEid**: The navigation node element identifier of the parent navigation node.

@AddIfNotThere: A bit representing whether to add a new navigation node if no corresponding navigation node exists. This value MUST be 0.

@UseRootFolderForNav: A bit representing whether to point the navigation node to the default list view of the list or to the lists's root folder. This value MUST be 0.

Return Values: proc_UpdateListNavNode returns an integer return code which MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.4.60   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
);
```

@EventTime: The UTC date time of the event relative to which the subscriptions are updated**.**

Return Values: Proc_UpdateSchedSubscriptionTimes returns an integer which MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.5 Timer Events

If the connection timeout event is triggered, the stored procedure is terminated and the call fails.

3.1.6 Other Local Events

No other local events 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 Message Processing Events and Sequencing Rules

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 will be 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
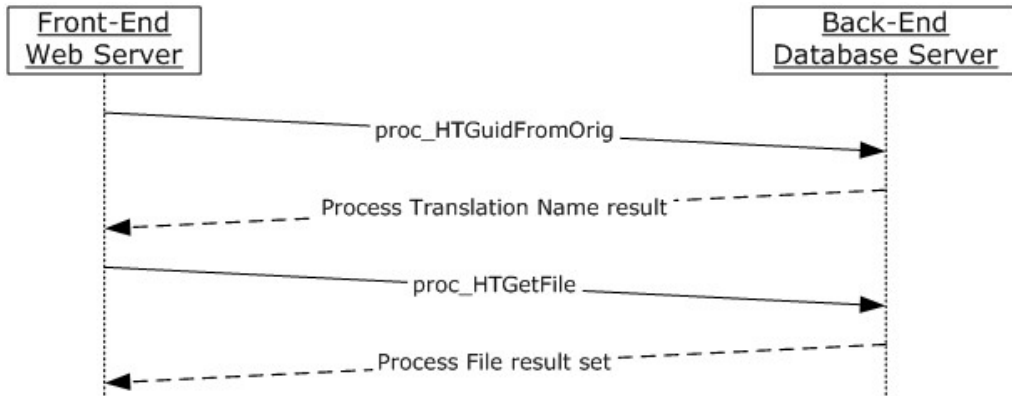
Figure 2: Retrieve the HTML translate cache item

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
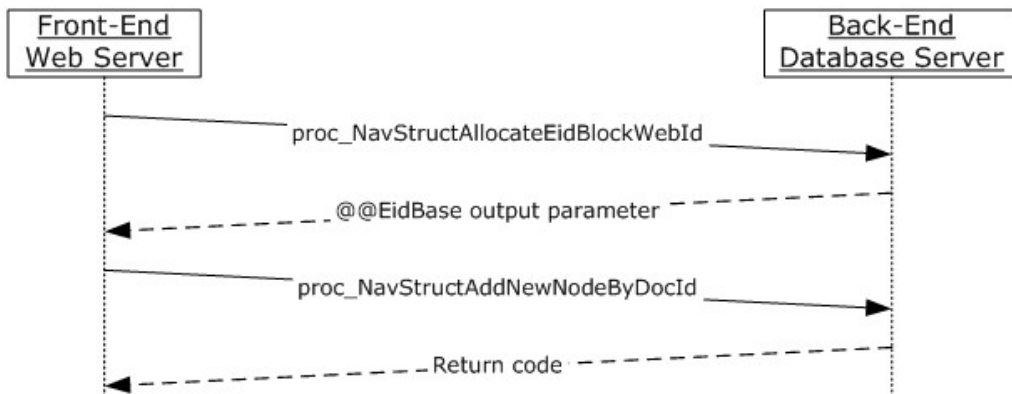
Figure 3: CreateNavigationNodeByDocId operation

CreateNavigationNodeByDocId gets invoked for creating a navigation node that points to a SharePoint object inside the site. 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
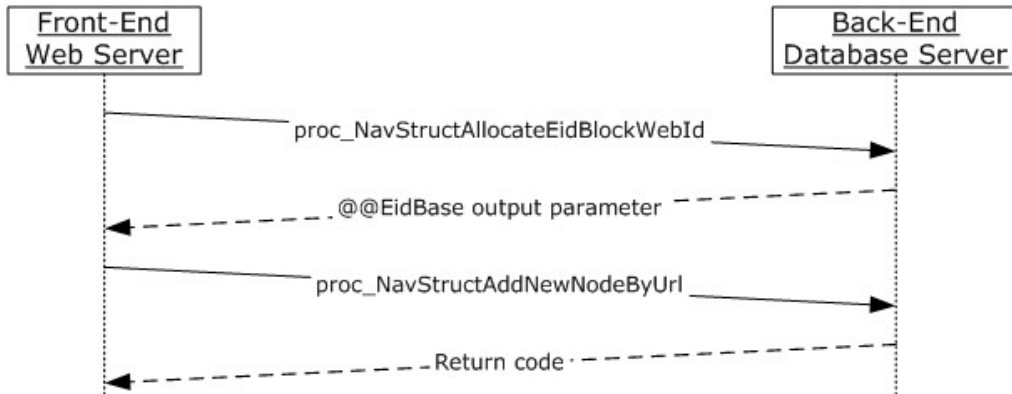
Figure 4: CreateNavigationNodeByURL operation

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* obtained in step 2.

3.2.4.4   AddAndUpdate NavigationNode
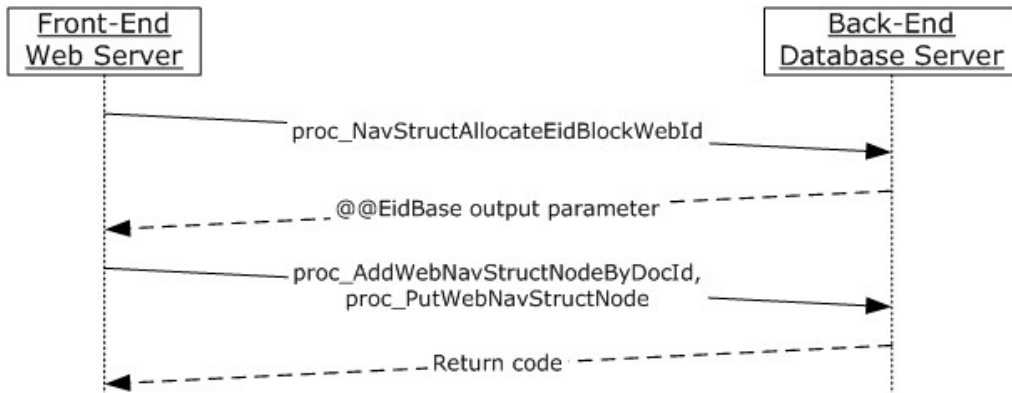
Figure 5: AddAndUpdateNavigationNode operation

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* obtained in step 2.

3.2.4.5 MoveNavigationNode
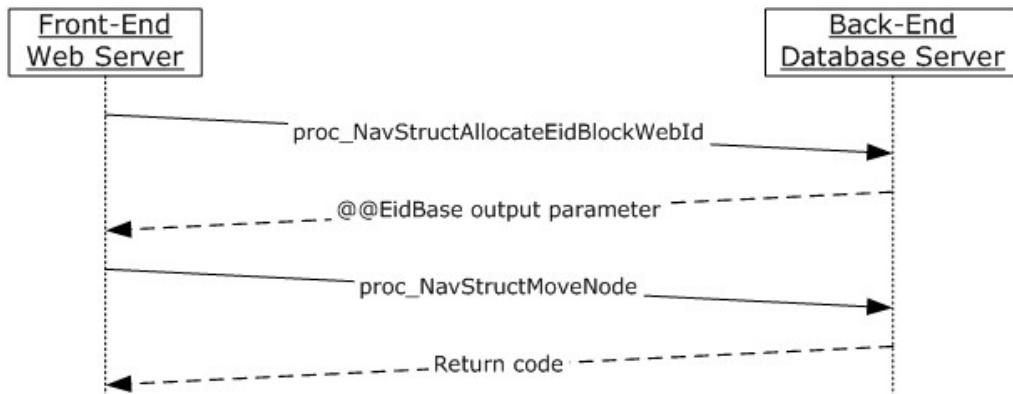
Figure 6: MoveNavigationNode operation

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* obtained in step 2.

3.2.5 Timer Events

If the connection timeout event is triggered, the connection and the stored procedure call fails.

3.2.6 Other Local Events

No other local events impact the operation of this protocol.

4   Protocol Examples

4.1   TreeView Operations

This example describes the requests made and responses returned between front-end Web server and back-end database server when an end user expands tree nodes on the TreeView (Section 3.1.1.5) of a site.

In this scenario, the site contains a SubSite and a document library containing folders.

1. When the client requests the information for the site, the front-end Web server gets the information of the SubSites and the lists in the site by calling proc_GetWebAndChildrenNSInfo (Section 3.1.4.25) to the back-end database server.

2. Back-end database server returns one SiteNSInfo result set containing the SubSites of the expanded site node and one ListNSInfoResultSet containing document libraries and lists under the expanded site node.

3. When the client requests information about the document library Node under the expanded site node, the front-end Web server gets the information of the document library and its subfolders by calling proc_GetWebAndChildrenNSInfo (Section 3.1.4.25) 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 previous call 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 the previous comment, the front-end Web server calls **proc_AddDocComment** with the comment id 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**.

Release: Sunday, December 19, 2010

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 **site-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 by calling **proc_GetWebNavStruct**.

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 user 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-WSSFO] 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® SQL Server® 2005

- Microsoft® SQL Server® 2008

- Microsoft® SQL Server® 2008 R2

- Windows® SharePoint® Services 3.0

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 2.2.2.3: Windows SharePoint Services 3.0 treats the values of 2 and 4 in the same manner

<2> Section 2.2.6.3: 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 3.1.4.1: 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 0.

<4> Section 3.1.4.1.1: 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 0 and the result set contains the successfully inserted values. If the server fails to add the comment, then the server returns a nonzero error code and the result set contains the values that it tried to insert in the final attempt.

<5> Section 3.1.4.28.3: 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 3.1.4.34: The client in Windows SharePoint Services 3.0 creates a unique Translation Name for each file. Thus there is only one row corresponding to a SiteID, DirName, LeafName in the table.

<7> Section 3.1.4.36.1: The client in Windows SharePoint Services 3.0 uses only the first row that is returned in the result set.

<8> Section 3.1.4.42: Windows SharePoint Services 3.0 does not honor this parameter. It hides the Navigation Nodes based on the permissions of the user.

<9> Section 3.1.4.43: 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 0.

<10> Section 3.1.4.49: Windows SharePoint Services 3.0 SP1 always returns an error token [MS-TDS], indicating a syntax error.

<11> Section 3.1.4.51: 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 0.

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