

[MS-WSSDLIM2]: Windows SharePoint Services: Content Database Document and List Item Management Communications Version 2 Protocol Specification

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

Tools. The Open Specifications do not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them. Certain Open Specifications are intended for use in conjunction with publicly available standard

specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

Revision Summary

Date	Revision History	Revision Class	Comments
07/13/2009	0.1	Major	Initial Availability
08/28/2009	0.2	Editorial	Revised and edited the technical content
11/06/2009	0.3	Editorial	Revised and edited the technical content
02/19/2010	1.0	Major	Updated and revised the technical content
03/31/2010	1.01	Editorial	Revised and edited the technical content
04/30/2010	1.02	Editorial	Revised and edited the technical content
06/07/2010	1.03	Editorial	Revised and edited the technical content
06/29/2010	1.04	Minor	Clarified the meaning of the technical content.
07/23/2010	1.05	Editorial	Changed language and formatting in the technical content.
09/27/2010	1.05	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	1.06	Major	Significantly changed the technical content.
12/17/2010	1.07	Editorial	Changed language and formatting in the technical content.
03/18/2011	1.7.1	Editorial	Changed language and formatting in the technical content.
06/10/2011	1.7.1	No change	No changes to the meaning, language, or formatting of the technical content.
01/20/2012	1.7.1	No change	No changes to the meaning, language, or formatting of the technical content.
04/11/2012	1.7.1	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	1.7.1	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction	10
1.1 Glossary	10
1.2 References	14
1.2.1 Normative References	15
1.2.2 Informative References	15
1.3 Protocol Overview (Synopsis)	16
1.3.1 Change Log Operations	16
1.3.2 Publish and Un-publish Operations	16
1.3.3 Check-In and Check-Out Operations	16
1.3.4 Historical Versioning Operations	16
1.3.5 Link Fixup Operations	16
1.3.6 Lookup Relationships Operations	16
1.3.7 Theme Operations	16
1.3.8 Wide List Operations	16
1.3.9 File Fragment Operations	17
1.4 Relationship to Other Protocols	17
1.5 Prerequisites/Preconditions	17
1.6 Applicability Statement	17
1.7 Versioning and Capability Negotiation	17
1.8 Vendor-Extensible Fields	17
1.9 Standards Assignments	17
2 Messages	18
2.1 Transport	18
2.2 Common Data Types	18
2.2.1 Simple Data Types and Enumerations	18
2.2.2 Simple Data Types	18
2.2.2.1 Change Log ListId	18
2.2.2.2 Change Log ItemId	21
2.2.2.3 Change Log DocId	23
2.2.2.4 Change Log Guid0	24
2.2.2.5 Change Log Int0	27
2.2.2.6 Change Log ContentTypeId	30
2.2.2.7 Change Log ItemFullUrl	31
2.2.2.8 Change Log TimeLastModified	33
2.2.2.9 Change Log ItemName	36
2.2.2.10 Change Log Int1	38
2.2.2.11 Change Log SiteId	41
2.2.2.12 Change Log WebId	42
2.2.3 Bit Fields and Flag Structures	42
2.2.3.1 Event Object Type Flags	42
2.2.3.2 Event Type Flags	43
2.2.3.3 Security Change Type Flags	44
2.2.3.4 Delete Flags	44
2.2.3.5 Document Flags	44
2.2.4 Enumerations	46
2.2.4.1 Relationship Delete Behavior Type	46
2.2.5 Binary Structures	46
2.2.6 Result Sets	46
2.2.6.1 Site Collection Flags Result Set	46

2.2.6.2	Distribution List E-mail Address Result Set.....	46
2.2.6.3	Document Content Stream Transfer Result Set.....	47
2.2.6.4	Document Version Content Stream Transfer Result Set	47
2.2.6.5	Document Stream Storage Migration Result Set	48
2.2.6.6	Document Version Stream Storage Migration Result Set	49
2.2.6.7	File Fragment Result Set.....	49
2.2.7	Tables and Views	50
2.2.7.1	AllUserData Table	50
2.2.7.2	NameValuePair Table	50
2.2.7.3	NameValuePair_Latin1_General_CI_AS Table	51
2.2.7.4	Collated NameValuePair Tables.....	51
2.2.8	XML Structures	51
2.2.8.1	Namespaces	51
2.2.8.2	Simple Types	51
2.2.8.3	Complex Types.....	52
2.2.8.3.1	List Data Source	52
2.2.8.4	Elements	52
2.2.8.5	Attributes	52
2.2.8.6	Groups	52
2.2.8.7	Attribute Groups.....	52

3 Protocol Details..... 53

3.1	Server Details	53
3.1.1	Abstract Data Model	53
3.1.1.1	Change Log Operations	53
3.1.1.2	Publish and Un-publish Operations.....	53
3.1.1.3	Check-In and Check-Out Operations	53
3.1.1.4	Historical Versioning Operations	54
3.1.1.5	Link Fixup Operations.....	54
3.1.1.6	Lookup Relationships Operations	55
3.1.1.7	Theme Operations	55
3.1.1.8	Wide List Operations	57
3.1.1.9	File Fragment Operations.....	57
3.1.2	Timers	57
3.1.3	Initialization	57
3.1.4	Message Processing Events and Sequencing Rules.....	57
3.1.4.1	fn_RoundDateToNearestSecond.....	58
3.1.4.2	fn_UnpackCsvString.....	58
3.1.4.3	proc_AddDependency.....	58
3.1.4.4	proc_AddEventToCache	59
3.1.4.5	proc_AddGhostDocument	60
3.1.4.6	proc_AddNewRowOrdToList.....	63
3.1.4.7	proc_AddNewRowOrdToListItem	64
3.1.4.8	proc_AL.....	65
3.1.4.9	proc_AppendFileFragmentDataById	67
3.1.4.10	proc_CascadeDeleteItems.....	68
3.1.4.10.1	Deleted Documents Result Set	71
3.1.4.11	proc_CheckIfExistingFieldHasDuplicateValues	71
3.1.4.12	proc_CheckIfExistingLookupsHaveValidParents	72
3.1.4.13	proc_CheckIfRestoreNeedsTranLock	73
3.1.4.14	proc_CheckoutDocumentInternal	74
3.1.4.15	proc_CloneDoc	77
3.1.4.16	proc_ConvertJunctionToLookup	78

3.1.4.17	proc_ConvertLookupToJunction	79
3.1.4.18	proc_CopyUrl	79
3.1.4.18.1	NULL List Metadata Result Set	83
3.1.4.18.2	Copied Directory Result Set	84
3.1.4.19	proc_CreateList	84
3.1.4.19.1	List Metadata Result Set	88
3.1.4.19.2	Id and Full URL Result Set	88
3.1.4.20	proc_CreateSharedAccessRequest	88
3.1.4.21	proc_CreateSite	88
3.1.4.21.1	Site Owner Audit Mask Result Set	91
3.1.4.21.2	Site Secondary Contact Audit Mask Result Set	91
3.1.4.21.3	Site Administrator Audit Mask Result Set	91
3.1.4.21.4	Site Author Audit Mask Result Set	91
3.1.4.21.5	Site Contributor Audit Mask Result Set	91
3.1.4.21.6	Site Browser Audit Mask Result Set	91
3.1.4.21.7	Site Guest Audit Mask Result Set	92
3.1.4.22	proc_CreateView	92
3.1.4.23	proc_CreateWeb	93
3.1.4.23.1	Audit Flags Result Set	96
3.1.4.24	proc_DeleteAllItemVersions	97
3.1.4.25	proc_DeleteAttachment	98
3.1.4.26	proc_DeleteAttachmentsFolder	98
3.1.4.27	proc_DeleteChanges	99
3.1.4.28	proc_DeleteEventLog	100
3.1.4.29	proc_DeleteFileFragmentsByTag	100
3.1.4.30	proc_DeleteItemVersion	101
3.1.4.31	proc_DeleteSite	102
3.1.4.31.1	Site Collection Flags Result Set	103
3.1.4.31.2	Distribution List E-mail Address Result Set	103
3.1.4.32	proc_DeleteSiteAsync	103
3.1.4.32.1	Site Collection Flags Result Set	104
3.1.4.32.2	Distribution List E-mail Address Result Set	104
3.1.4.33	proc_DeleteSiteCoreAsync	104
3.1.4.34	proc_RestoreSite	104
3.1.4.35	proc_DeleteView	105
3.1.4.36	proc_DeleteWeb	106
3.1.4.36.1	Audit Flags Result Set	108
3.1.4.37	proc_RecycleWeb	108
3.1.4.37.1	Audit Flags Result Set	109
3.1.4.38	proc_DirtyDocWithForwardLinks	110
3.1.4.39	proc_DropListRecord	110
3.1.4.40	proc_DropListUniqueField	112
3.1.4.41	proc_DropLookupRelationship	113
3.1.4.42	proc_EnsureTranLockNotRequired	114
3.1.4.43	proc_FetchFileFormatMetaInfo	114
3.1.4.43.1	FileFormatMetaInfo Rowset	115
3.1.4.44	proc_FetchOldDoc	115
3.1.4.44.1	Domain Group Cache Versions Result Set	117
3.1.4.44.2	Domain Group Cache Back-End Database Server Update Result Set	117
3.1.4.44.3	Domain Group Cache Front-End Web Server Update Result Set	118
3.1.4.44.4	Document Version Metadata Result Set	118
3.1.4.44.5	Document Version Content Stream Result Set	121
3.1.4.44.6	Site Metadata Result Set	123

3.1.4.44.7	Event Receivers Result Set	123
3.1.4.44.8	Web Event Receivers Result Set	123
3.1.4.44.9	Site Features List Result Set	123
3.1.4.44.10	Web Parts Metadata, Nonpersonalized Result Set.....	123
3.1.4.44.11	List Metadata Result Set	124
3.1.4.44.12	List Event Receivers Result Set.....	124
3.1.4.44.13	List Security Information Result Set	124
3.1.4.44.14	Site Collection Custom Actions Result Set	124
3.1.4.44.15	Site Custom Actions Result Set.....	125
3.1.4.44.16	List Custom Actions Result Set	125
3.1.4.44.17	List Web Parts Result Set.....	125
3.1.4.44.18	NULL Content Type Order Result Set	125
3.1.4.44.19	Navigation Context Security Information Result Set	126
3.1.4.44.20	NULL Navigation Context Security Information Result Set	126
3.1.4.44.21	Empty Navigation Context Security Information Result Set	126
3.1.4.45	proc_FileFragmentPermissionCheck.....	126
3.1.4.46	proc_FindDocs.....	127
3.1.4.46.1	Found Docs Result Set	129
3.1.4.47	proc_FinishUndirtyList	129
3.1.4.48	proc_GenerateUniqueFileName	129
3.1.4.48.1	Unique File Name Result Set	130
3.1.4.49	proc_GetAllAttachmentsInfo	130
3.1.4.49.1	List Attachments Result Set	131
3.1.4.49.2	Document Attachments Result Set.....	131
3.1.4.50	proc_GetAllListsPlusProperties	131
3.1.4.50.1	AllListsPlus Properties Result Set	132
3.1.4.51	proc_GetChanges	132
3.1.4.51.1	EventInformation Result Set	133
3.1.4.51.2	EventDetails Result Set	134
3.1.4.52	proc_GetChangeToken	135
3.1.4.52.1	Change Token Result Set.....	135
3.1.4.52.2	Change Token Time Result Set.....	135
3.1.4.53	proc_GetCurrent	135
3.1.4.53.1	EventInformation Result Set	136
3.1.4.54	proc_GetDocIdUrl	136
3.1.4.55	proc_GetFileFragmentsById	137
3.1.4.56	proc_GetFileFragmentsByTag	138
3.1.4.57	proc_GetFullLinkInfoForSingleDoc	139
3.1.4.57.1	Web List For Normalization Result Set	140
3.1.4.57.2	NULL Individual URL Security Result Set.....	140
3.1.4.57.3	Individual URL Security Result Set.....	141
3.1.4.57.4	Document Link Information Result Set	141
3.1.4.57.5	Document Setup Path Result Set	142
3.1.4.58	proc_GetListDataLinks	142
3.1.4.58.1	Web List For Normalization Result Set	143
3.1.4.58.2	List Data Link Information Result Set	144
3.1.4.59	proc_GetListDataSource	146
3.1.4.59.1	List Data Source Result Set	146
3.1.4.60	proc_GetListItemsTreeToDelete	147
3.1.4.60.1	List Items Tree Result Set	148
3.1.4.60.2	Restrict Delete Error Result Set	149
3.1.4.61	proc_GetNewListItemId	149
3.1.4.62	proc_GetRelatedFieldsForList.....	150

3.1.4.62.1	List Related Fields Result Set	150
3.1.4.63	proc_GetSiteDeletionBatch.....	151
3.1.4.63.1	Site Deletion Batch Result Set.....	151
3.1.4.64	proc_GetUrlDocId	152
3.1.4.64.1	Directory And Leaf Names Result Set	152
3.1.4.65	proc_GetWebUrlFromId	152
3.1.4.66	proc_InsertEventSubscriptionJunctionEntries	153
3.1.4.67	proc_InsertItemIntoNameValuePair.....	166
3.1.4.68	proc_InsertItemIntoNameValuePairCollated	168
3.1.4.69	proc_InsertJunction.....	169
3.1.4.70	proc_InsertListUniqueField	170
3.1.4.71	proc_InsertLookupRelationship	170
3.1.4.72	proc_IsFieldALookupRelationship	171
3.1.4.73	proc_IsSiteScheduledForDeletion	172
3.1.4.74	proc_ListThemeFiles.....	172
3.1.4.74.1	Theme Files Information Result Set	173
3.1.4.75	proc_ListThemes.....	174
3.1.4.75.1	Theme Information Result Set.....	174
3.1.4.76	proc_LoadTheme	175
3.1.4.76.1	Theme Files Information Result Set	177
3.1.4.76.2	Theme INF File Information Result Set	178
3.1.4.77	proc_LogChange	178
3.1.4.78	proc_LogChangeForFileFragments.....	179
3.1.4.79	proc_PatchLinkForFile.....	180
3.1.4.80	proc_PatchLinkForWeb	181
3.1.4.81	proc_ReadSharedAccessRequests	182
3.1.4.81.1	Shared Access Requests Result Set.....	182
3.1.4.82	proc_RefreshCheckout.....	183
3.1.4.82.1	Document Metadata Result Set	183
3.1.4.82.2	NULL Result Set	183
3.1.4.83	proc_RemoveJunctions	184
3.1.4.84	proc_RemoveSharedAccessRequest.....	184
3.1.4.85	proc_RenameHostHeaderSite	185
3.1.4.86	proc_RenameSite.....	185
3.1.4.87	proc_SaveFileFormatMetaInfo	186
3.1.4.88	proc_SaveFileFragmentById	187
3.1.4.89	proc_SaveFileFragmentByTag.....	188
3.1.4.90	proc_SetListDataSource.....	189
3.1.4.91	proc_SetNextId	190
3.1.4.92	proc_StartUndirtyList	190
3.1.4.92.1	Cache Parse Identifier Result Set.....	191
3.1.4.93	proc_TakeOfflineDocument	191
3.1.4.93.1	Document Metadata Result Set	192
3.1.4.93.2	Event Receivers Result Set	192
3.1.4.93.3	NULL Result Set	193
3.1.4.93.4	Link Info Single Doc Result Set	193
3.1.4.94	proc_UndirtyListItem.....	193
3.1.4.95	proc_UpdateEntityIdForList	194
3.1.4.96	proc_TransExtendLockWeb	194
3.1.4.97	proc_TransLockWeb.....	195
3.1.4.98	proc_TransUnlockWeb.....	195
3.1.4.99	proc_UpdateDirtyDocument	196
3.1.4.100	proc_UpdateItemInNameValuePair	197

3.1.4.101	proc_UpdateItemInNameValuePairCollated	199
3.1.4.102	proc_UpdateLookupRelationship	200
3.1.4.103	proc_UpdateOrderNumber	201
3.1.4.104	proc_UpdateVersionVirusInfo	202
3.1.4.105	proc_UpdateView	203
3.1.4.106	proc_UpdateVirusInfo	205
3.1.4.107	proc_UpdateWebPartLinks	206
3.1.4.108	proc_UserHasDataItems	207
3.1.4.109	proc_ValidateLookupParents	208
3.1.4.110	proc_ValidateUniqueFields	210
3.1.4.111	proc_ValidateUniqueFieldsCollated	212
3.1.4.112	proc_VerifyListItemsTreeToDelete	214
3.1.4.113	proc_DeleteFileFragmentsById	215
3.1.4.114	proc_DeleteFileFragmentsForCleanup	216
3.1.4.115	proc_DeleteFileFragmentsForUserDataDelete	217
3.1.4.116	proc_GetDocStreamsForExternalMigration	218
3.1.4.117	proc_GetDocStreamsForInlineMigration	218
3.1.4.118	proc_GetDocStreamsForRbsMigration	219
3.1.4.118.1	Remote Blob Storage Document Stream Migration Result Set	220
3.1.4.119	proc_GetDocVersionStreamsForExternalMigration	221
3.1.4.120	proc_GetDocVersionStreamsForInlineMigration	222
3.1.4.121	proc_GetDocVersionStreamsForRbsMigration	222
3.1.4.121.1	Remote Blob Storage Document Version Stream Migration Result Set	223
3.1.4.122	proc_GetStreamsForBackup	224
3.1.4.122.1	Document Content Stream Backup Result Set	224
3.1.4.122.2	Document Version Content Stream Backup Result Set	224
3.1.4.123	proc_GetStreamsForSiteMove	224
3.1.4.123.1	Document Content Stream Site Move Result Set	225
3.1.4.123.2	Document Version Content Stream Site Move Result Set	225
3.1.4.124	proc_EnsureSiteRbsCollection	225
3.1.4.125	proc_GetStreamsForPreSiteMove	226
3.1.4.125.1	Remote Blob Collection Identifier Preliminary Site Move Result Set	226
3.1.4.125.2	Document Content Preliminary Site Move Result Set	226
3.1.4.125.3	Document Version Content Preliminary Site Move Result Set	226
3.1.4.126	proc_ListRbsStoresWithIds	226
3.1.4.126.1	List Remote Blob Storage Stores With Identifiers Result Set	227
3.1.4.127	proc_ShallowCopyRbsBlobs	227
3.1.5	Timer Events	228
3.1.6	Other Local Events	228
3.2	Client Details	228
3.2.1	Abstract Data Model	228
3.2.2	Timers	228
3.2.3	Initialization	228
3.2.4	Message Processing Events and Sequencing Rules	229
3.2.5	Timer Events	229
3.2.6	Other Local Events	229
4	Protocol Examples	230
4.1	Change Log	230
4.2	Link Fixup	231
4.3	Themes	233
4.4	Add Just Enough Fields to Cause Allocation of Additional Rows Per List Item	234
4.5	Allocate New Rows While Inserting an Item into a Wide List	235

4.6 Cascade Delete Relationship Behavior	236
4.7 File Fragment Update Behavior	237
5 Security	238
5.1 Security Considerations for Implementers	238
5.2 Index of Security Parameters	238
6 Appendix A: Product Behavior	239
7 Change Tracking.....	240
8 Index	241

1 Introduction

This document specifies the Windows SharePoint Services: Content Database Document and List Item Management Communications Version 2 Protocol, which specifies the communication sequences used by the front-end Web server and application servers to perform data query and update commands on back-end database servers as part of document and list item management operations.

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

1.1 Glossary

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

- access control list (ACL)**
- access mask**
- anonymous user**
- application**
- ASCII**
- Component Object Model (COM)**
- Coordinated Universal Time (UTC)**
- domain**
- file system**
- GUID**
- Hypertext Transfer Protocol (HTTP)**
- Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)**
- language code identifier (LCID)**
- security policy**
- Unicode**
- XML**

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

- 12-hour clock notation**
- 24-hour clock notation**
- alert**
- alert subscription**
- assembly**
- assembly name**
- attachment**
- attachment identifier**
- audit entry**
- audit flag**
- author**
- back-end database server**
- backward link**
- base type**
- base view identifier**
- binary large object (BLOB)**
- Boolean**
- bot**
- calendar type**

cascading behavior
cascading style sheet (CSS)
change log identifier
check in
check out
checked out
class identifier (CLSID)
co-authoring transition
Collaborative Application Markup Language (CAML)
collation
collation identifier
collation order
column
configuration database
content database
content type
content type identifier
CSS
current user
current version
custom action
customized
data source
datetime
delete flag
delete transaction identifier
directory name
dirty
display name
displayed version
distribution list
document
document flag
document identifier
document library
document store type
document stream
document version
domain group
draft
dynamic page
dynamic Web template
editor
e-mail address
empty GUID
Entity
event
event handler
event object type flag
event receiver
event sink
event type flag
external group
farm
feature identifier

field
field definition
field identifier
file
file fragment
file fragment identifier
file fragment partition
file fragment tag
fixed schema
folder
form digest validation
Forms folder
forward link
front-end Web server
full URL
fully qualified URL
function
group
historical version
history link
host header
HTTP entity tag
HTTP GET
HTTP HEAD
hyperlink
Hypertext Markup Language (HTML)
indexed field
internal version number
Internet Information Services (IIS)
item
item identifier
item order field
leaf name
level
link bar
link fixup
list
list identifier
list item
list item attachment
list item identifier
list schema
list server template
list template
list view
locked
login name
lookup field
major version
master page
meeting instance
Meeting Workspace site
member
metadict
minor version

moderated object
multilingual user interface (MUI)
multivalued lookup field
navigation structure
page type
parent site
partition
path segment
permission
permission level
personal view
policy
provisioned
published
published version
publishing level
query
read-only mode
Recycle Bin
Recycle Bin item
relationship delete behavior
relationship lookup field
request identifier
resource token
restrict behavior
result set
return code
rights
role assignment
role definition
root folder
row
row ordinal
security group
security principal
security provider
security scope
server-relative URL
shared view
site
site collection
site collection administrator
site collection flag
site collection identifier
site collection quota
site column
site definition
site definition version
site identifier
site-relative URL
static page
stored procedure
store-relative form
store-relative URL
subsite

SystemID
theme
thicket
thicket folder
thicket supporting file
top-level site
transaction application lock
transaction identifier
Transact-Structured Query Language (T-SQL)
uncustomized
Uniform Resource Locator (URL)
unique column constraint
URL zone
user identifier
user interface (UI) version
version
view
View
view flag
view identifier
virus scanner
Web application
Web bot
Web Part
Web Part identifier
Web Part Page
Web Part property
Web Part type identifier
Web Part zone
Web Part zone identifier
Welcome page
wide list
XML namespace
XML schema
XML schema definition (XSD)
zero-based index

The following terms are specific to this document:

calculated field: A user-defined field that can perform calculations by using the contents of other fields.

list flag: An 8-byte unsigned integer bit mask that provides metadata about a SharePoint list.

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

1.2 References

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

1.2.1 Normative References

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

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

[MC-FPSEWM] Microsoft Corporation, "[FrontPage Server Extensions: Website Management Specification](#)".

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

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

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

[MS-WPPS] Microsoft Corporation, "[Web Part Pages Web Service Protocol Specification](#)".

[MS-WSSCAML] Microsoft Corporation, "[Collaborative Application Markup Language \(CAML\) Structure Specification](#)".

[MS-WSSEUX2] Microsoft Corporation, "[Windows SharePoint Services: Content Database End-User Experience Communications Version 2 Protocol Specification](#)".

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

[MS-WSSTS] Microsoft Corporation, "[Windows SharePoint Services Technical Specification](#)".

[RFC1950] Deutsch, P., and Gailly, J-L., "ZLIB Compressed Data Format Specification version 3.3", RFC 1950, May 1996, <http://www.ietf.org/rfc/rfc1950.txt>

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

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

1.2.2 Informative References

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

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

[MS-WSSCCSP] Microsoft Corporation, "[Windows SharePoint Services: Content Database Core List Schema and Site Provisioning Communications Protocol Specification](#)".

[MS-WSSCCSP2] Microsoft Corporation, "[Windows SharePoint Services Content Database Core List Schema and Site Provisioning Communications Version 2 Protocol Specification](#)".

[MS-WSSO] Microsoft Corporation, "[Windows SharePoint Services Overview](#)".

1.3 Protocol Overview (Synopsis)

This protocol specifies the communication between the **front-end Web server** and the **back-end database server** used to satisfy requests involving the following operations:

1.3.1 Change Log Operations

Includes methods to retrieve or append entries to the change log. The Change Log contains information about **item** actions such as Add, Update, Delete, Rename, Move Away, and Move Into, at the **list (1)**, **site (2)**, **site collection**, and **content database** levels. An **application** could use this information to find out what changes have occurred on the data objects stored in the back-end database server. It could further use this information to implement synchronization features by replaying these events on a different site.

1.3.2 Publish and Un-publish Operations

Includes methods to change the **publishing level** of a document in a list.

1.3.3 Check-In and Check-Out Operations

Includes methods to **check out** and **check in** a document in a list.

1.3.4 Historical Versioning Operations

Includes methods for managing the **historical versions** of a document or list item.

1.3.5 Link Fixup Operations

A back-end database server and front-end Web server work together to implement **link fixup**. The back-end database server tracks **forward links** from list items to documents. When the back-end database server performs an operation that requires link fixup for list items, it defers potentially complex work by marking the list items needing link fixup as **dirty**. Later, before a front-end Web server retrieves list data, it checks the "list is dirty" status and, if the list is dirty, performs a link fixup operation before querying the data in the list.

1.3.6 Lookup Relationships Operations

Includes methods to support relationships between lists (1) and the cascading behavior and **restrict behavior** when deleting **list items** in these lists (1).

1.3.7 Theme Operations

Includes methods to retrieve **theme** information for the purposes of applying a theme to the pages that belong to a site.

1.3.8 Wide List Operations

Includes methods to create an unlimited number of fields for lists. The content database for a back-end database server has a **fixed schema**. To accommodate for this, the concept of **wide list** and **row ordinal** were created.

If the number of fields in a list of a particular type exceeds a fixed maximum for that type per row in the content database, then new **rows** are allocated for every list item in the list.

1.3.9 File Fragment Operations

Includes methods to add, update, delete, and retrieve **file fragments** associated with a **document**. These objects are opaque implementation-specific data used by applications to enable additional features related to document editing.

1.4 Relationship to Other Protocols

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

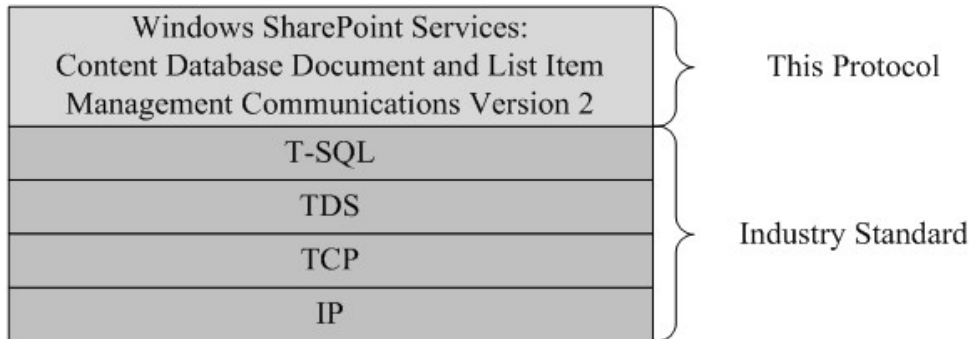


Figure 1: This protocol in relation to other protocols

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 described 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 views or 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 Change Log ListId

Change log **ListId** is a **GUID** and is part of a change log entry. This data has different meanings based on the event object type and event type data of the same change log entry. The possible meanings of this value are specified as the following:

Event Object Type = 0x00000001 (list item)

Event type	Description
0x00000010	The identifier of the list (1) that contains the list item.
0x00000020	The identifier of the list (1) that contains the list item.
0x00000040	The identifier of the list (1) that contains the list item.
0x00000080	The identifier of the list (1) that contains the list item.
0x00000100	The identifier of the list (1) that contains the list item.
0x00001001	The identifier of the list (1) that contains the list item.
0x00002000	The identifier of the list (1) that contains the list item.
0x00002002	The identifier of the list (1) that contains the list item.
0x00004004 or 0x00004000	The identifier of the list (1) that contains the list item.
0x00008000	The identifier of the list (1).
0x00010000	The identifier of the list (1) to which this list item was moved.
0x00020009 or 0x00020000	The identifier of the list (1) that contains the list item.
0x00080000	The identifier of the list (1) that contains the list item.
0x00100000	The identifier of the list (1) that contains the list item.
0x02000000	The identifier of the list (1) that contains the list item.
0x04000000	The identifier of the list (1) that contains the list item.

Event Object Type = 0x00000002 (List (1))

Event type	Description
0x00001000	The identifier of the list (1).
0x00002000	The identifier of the list (1).
0x00004000	The identifier of the list (1).
0x00008000	The identifier of the list (1).
0x00020000	The identifier of the list (1).
0x00080000	The identifier of the list (1).
0x02000000	The identifier of the list (1).

Event Object Type = 0x00000004 (Site (2))

MUST be NULL.

Event Object Type = 0x00000008 (site collection)

MUST be NULL.

Event Object Type = 0x00000010 (**File**)

Event type	Description
0x00001000	MUST be NULL.
0x00004000	MUST be NULL.
0x00010000	The identifier of the list (1) into which the file is being moved.
0x00020000	MUST be NULL.
0x00008000	MUST be NULL.
0x00100000	MUST be NULL.
0x04000000	The identifier of the list (1) from which the file is being moved.

Event Object Type = 0x00000020 (**Folder**)

Event type	Description
0x00001000	MUST be NULL.
0x00002000	MUST be NULL.
	MUST be NULL.

Event type	Description
0x00004000	
0x00008000	MUST be NULL.
0x00010000	The identifier of the list (1) into which the folder is moved.
0x00020000	MUST be NULL.
0x00100000	MUST be NULL.
0x04000000	The identifier of the list which the folder used to belong.

Event Object Type = 0x00000040 (**alert subscription**)

Event type	Description
0x00001000	The identifier of the list (1) that contains this alert subscription
0x00002000	The identifier of the list (1) that contains this alert subscription.
0x00004000	The identifier of the list (1) that contains this alert subscription.

Event Object Type = 0x00000080 (**security principal (2)**)

MUST be NULL.

Event Object Type = 0x00000100 (**security group**)

MUST be NULL.

Event Object Type = 0x00000200 (**Content type**)

MUST be NULL.

Event Object Type = 0x00000400 (**site column**)

MUST be NULL.

Event Object Type = 0x00000800 (**Security policy**)

MUST be NULL.

Event Object Type = 0x00001000 (View)

Event type	Description
0x00001000	The identifier of the list (1) that contains the view .
0x00002000	The identifier of the list (1) that contains the view.
0x00004000	The identifier of the list (1) that contains the view.

2.2.2.2 Change Log ItemId

Change log **ItemId** is an integer and is part of a change log entry. This data has different meanings based on the event object type and event type data of the same change log entry. The possible meanings of this value are specified as the following:

Event Object Type = 0x00000001 (list item)

Event type	Description
0x00000010	The item identifier of the list item.
0x00000020	The item identifier of the list item.
0x00000040	The item identifier of the list item.
0x00000080	The item identifier of the list item.
0x00000100	The item identifier of the list item.
0x00001001	The item identifier of the list item.
0x00002000	The item identifier of the list item.
0x00002002	The item identifier of the list item.
0x00004004	The item identifier of the list item.
0x00008000	The item identifier of the list item.
0x00010000	The item identifier of the list item.
0x00020009	The item identifier of the list item.
0x00080000	The item identifier of the list item.
0x00100000	The item identifier of the list item.
0x02000000	The item identifier of the list item.
0x04000000	The old item identifier of the list item.

Event Object Type = 0x00000002 (List)

MUST be NULL.

Event Object Type = 0x00000004 (Site (2))

MUST be NULL.

Event Object Type = 0x00000008 (site collection)

MUST be NULL.

Event Object Type = 0x00000010 (File)

Event type	Description
0x00001000	MUST be NULL.
0x00004000	MUST be NULL.
0x00010000	The item identifier of the file inside the list which it is being moved into.
0x00020000	MUST be NULL.
0x00008000	MUST be NULL.
0x00100000	MUST be NULL.
0x04000000	The item identifier of the file inside the list which it is being moved away from.

Event Object Type = 0x00000020 (Folder)

Event Type	Description
0x00001000	MUST be NULL.
0x00002000	MUST be NULL.
0x00004000	MUST be NULL.
0x00008000	MUST be NULL.
0x00010000	The item identifier of the folder in the list (1) after the move.
0x00020000	MUST be NULL.
0x00100000	MUST be NULL.
0x04000000	The item identifier of the folder in the list before the move.

Event Object Type = 0x00000040 (alert subscription)

Event type	Description
0x00001000	The item identifier of the document, if this alert subscription is associated with this list item. If the alert subscription is associated with the list, this MUST be NULL.
0x00002000	The item identifier of the document.
0x00004000	The item identifier of the document.

Event Object Type = 0x00000080 (security principal (2))

Event type	Description
0x00001000	Equal to Int0 if the security principal (2) added is active; otherwise MUST be NULL.
0x00002000	SHOULD be the security principal (2) identifier, if the security principal (2) is active after the update, otherwise this MUST be NULL
0x00004000	MUST be NULL.

Event Object Type = 0x00000100 (security group)

Event type	Description
0x00001000	MUST be NULL.
0x00002000	MUST be NULL.
0x00004000	MUST be NULL.
0x00200000	The identifier of the security principal which is added to the security group.
0x00400000	The identifier of the security principal which is removed from the security group.

Event Object Type = 0x00000200 (Content type)

MUST be NULL.

Event Object Type = 0x00000400 (site column)

MUST be NULL.

Event Object Type = 0x00000800 (Security policy)

MUST be NULL.

Event Object Type = 0x00001000 (**View**)

MUST be NULL.

2.2.2.3 Change Log DocId

Change log **DocId** is a GUID and is part of a change log entry. This data has different meanings based on the event object type and event type data of the same change log entry. The possible meanings of this value are specified as the following:

Event Object Type = 0x00000001 (list item)

The **document identifier (2)** of the document associated with this list item.

Event Object Type = 0x00000002 (List (1))

MUST be NULL.

Event Object Type = 0x00000004 (Site (2))

MUST be NULL.

Event Object Type = 0x00000008 (site collection)

MUST be NULL.

Event Object Type = 0x00000010 (File)

The document identifier of the document.

Event Object Type = 0x00000020 (Folder)

The identifier of the folder.

Event Object Type = 0x00000040 (alert subscription)

MUST be NULL.

Event Object Type = 0x00000080 (security principal (2))

MUST be NULL.

Event Object Type = 0x00000100 (security group)

MUST be NULL.

Event Object Type = 0x00000200 (Content type)

MUST be NULL.

Event Object Type = 0x00000400 (site column)

MUST be NULL.

Event Object Type = 0x00000800 (Security policy)

MUST be NULL.

Event Object Type = 0x00001000 (View)

2.2.2.4 Change Log Guid0

Change log **Guid0** is a GUID and is part of a change log entry. This data has different meanings based on the event object type and event type data of the same change log entry. The possible meanings of this value are specified as the following:

Event Object Type = 0x00000001 (list item)

Event type	Description
0x00000010	MUST be NULL.
0x00000020	MUST be NULL.
0x00000040	MUST be NULL.
0x00000080	MUST be NULL.
0x00000100	MUST be NULL.
0x00001001	MUST be NULL.

Event type	Description
0x00002000	MUST be NULL.
0x00002002	MUST be NULL.
0x00004004	MUST be NULL.
0x00008000	MUST be NULL.
0x00010000	The identity of the list used to contain the list item.
0x00020009	MUST be NULL.
0x00080000	Identifier of the security scope where the role assignments are added.
0x00100000	MUST be NULL.
0x02000000	Identifier of the security scope where the role assignments are deleted.
0x04000000	The identifier of the list that this list item was moved to.

Event Object Type = 0x00000002 (List (1))

Event type	Description
0x00001000	MUST be NULL.
0x00002000	When a Field is deleted from the List (1), this value is the identifier of the Field. Otherwise, this value MUST be NULL.
0x00004000	MUST be NULL.
0x00008000	MUST be NULL.
0x00020000	MUST be NULL.
0x00080000	Identifier of the security scope where the role assignment is added.
0x00080000	Identifier of the security scope where the role assignments are added.
0x02000000	Identifier of the security scope where the role assignment is deleted.
0x02000000	Identifier of the security scope where the role assignments are deleted.

Event Object Type = 0x00000004 (Site (2))

Event type	Description
0x00001000	MUST be NULL.
0x00002000	MUST be NULL.
0x00004000	MUST be NULL.
0x00008000	MUST be NULL.
0x00040000	Identifier of the security scope where the role definition is added.

Event type	Description
0x00020000	MUST be NULL.
0x00800000	Identifier of the security scope where the role definition is deleted.
0x01000000	Identifier of the security scope where the role definition is modified.
0x02000000	Identifier of the security scope where the role assignment is deleted.
0x00080000	Identifier of the security scope where the role assignment is added.
0x08000000	MUST be NULL.

Event Object Type = 0x00000008 (site collection)

MUST be NULL.

Event Object Type = 0x00000010 (File)

MUST be NULL.

Event Object Type = 0x00000020 (Folder)

MUST be NULL.

Event Object Type = 0x00000040 (alert subscription)

Event Type	Description
0x00001000	The identifier of the alert subscription.
0x00002000	The identifier of the alert subscription.
0x00004000	The identifier of the alert subscription.

Event Object Type = 0x00000080 (security principal (2))

MUST be NULL.

Event Object Type = 0x00000100 (security group)

MUST be NULL.

Event Object Type = 0x00000200 (Content type)

MUST be NULL.

Event Object Type = 0x00000400 (site column)

MUST be NULL.

Event Object Type = 0x00000800 (Security policy)

MUST be NULL.

Event Object Type = 0x00001000 (View)

Event Type	Description
0x00001000	The identifier of the list view .
0x00002000	The identifier of the alert subscription.
0x00004000	The identifier of the alert subscription.

2.2.2.5 Change Log Int0

Change log **Int0** is an integer and is part of a change log entry. This data has different meanings based on the event object type and event type data of the same change log entry. The possible meanings of this value are specified as the following:

Event Object Type = 0x00000001 (list item)

Event type	Description
0x00000010	MUST be NULL.
0x00000020	MUST be NULL.
0x00000040	MUST be NULL.
0x00000080	MUST be NULL.
0x00000100	MUST be NULL.
0x00001001	MUST be NULL.
0x00002000	MUST be NULL.
0x00002002	MUST be NULL.
0x00004004	MUST be NULL.
0x00008000	MUST be NULL.
0x00010000	Old Identifier of the List Item.
0x00020009	MUST be NULL.
0x00080000	If the Change Log Int1 of the same Change Log Entry ("Int1") is NULL, this value MUST be -1. If Int1 is greater than or equal to 0x40000000, this value is the Security Principal Identifier of the role assignment that is added. If Int1 is NOT NULL and less than 0x40000000, this value is the Security Principal Identifier which has made the role inheritance change.
0x00100000	MUST be NULL.
0x02000000	If the Change Log Int1 of the same Change Log Entry ("Int1") is NULL, this value is the Security Principal Identifier that is removed from all Roles. If Int1 is greater than or equal to 0x40000000, this value is the Security Principal Identifier that is removed from the Role specified by Int1. If Int1 is not NULL, and less than 0x40000000, this value MUST be NULL.

Event type	Description
0x04000000	Current Identifier of the List Item.

Event Object Type = 0x00000002 (List)

Event type	Description
0x00001000	MUST be NULL.
0x00002000	MUST be NULL.
0x00004000	MUST be NULL.
0x00008000	MUST be NULL.
0x00020000	MUST be NULL.
0x00080000	If the change log Int1 of the same change log entry ("Int1") is NULL, this value MUST be - 1. If Int1 is greater than or equal to 0x40000000, this value is the security principal identifier of the role assignment that is added. If Int1 is NOT NULL and less than 0x40000000, this value is the security principal identifier which has made the role inheritance change.
0x02000000	If the change log Int1 of the same change log entry ("Int1") is NULL, this value is the security principal identifier that is removed from all roles. If Int1 is greater than or equal to 0x40000000, this value is the security principal identifier that is removed from the role specified by Int1 . If Int1 is not NULL and is less than 0x40000000, this value MUST be NULL.

Event Object Type = 0x00000004 (Site)

Event type	Description
0x00001000	MUST be NULL.
0x00002000	MUST be NULL.
0x00004000	MUST be NULL.
0x00008000	MUST be NULL.
0x00040000	SHOULD be the Security Principal Identifier who is adding this role definition.
0x00020000	MUST be NULL.
0x00800000	MUST be NULL.
0x01000000	MUST be NULL.
0x02000000	If the change log Int1 of the same change log entry ("Int1") is NULL, this value is the security principal identifier that is removed from all roles. If Int1 is greater than or equal to 0x40000000, this value is the security principal identifier that is removed from the role specified by Int1 . If Int1 is not NULL, and less than 0x40000000, this value MUST be NULL.
0x00080000	If the change log Int1 of the same change log entry ("Int1") is NULL, this value MUST be - 1. If Int1 is greater than or equal to 0x40000000, this value is the security principal identifier of the role assignment that is added. If Int1 is NOT NULL and is less than

Event type	Description
	0x40000000, this value is the security principal identifier which has made the role inheritance change.
0x08000000	MUST be NULL.

Event Object Type = 0x00000008 (Site Collection)

MUST be NULL.

Event Object Type = 0x00000010 (File)

MUST be NULL.

Event Object Type = 0x00000020 (Folder)

MUST be NULL.

Event Object Type = 0x00000040 (Alert Subscription)

MUST be NULL.

Event Object Type = 0x00000080 (Security Principal)

Event type	Description
0x00001000	Identifier of the security principal added.
0x00002000	User identifier.
0x00004000	Security principal identifier.

Event Object Type = 0x00000100 (Security Group)

MUST be Group Identifier.

Event Object Type = 0x00000200 (Content Type)

MUST be NULL.

Event Object Type = 0x00000400 (site column)

MUST be NULL.

Event Object Type = 0x00000800 (Security Policy)

MUST be NULL.

Event Object Type = 0x00001000 (View)

MUST be NULL.

2.2.2.6 Change Log ContentTypeId

Change log **ContentTypeId** is a numeric string value of arbitrary but limited length. It is part of a change log entry. This data has different meanings based on the event object type and event type data of the same change log entry. The possible meanings of this value are specified as the following:

Event Object Type = 0x00000001 (List Item)

MUST be NULL.

Event Object Type = 0x00000002 (List)

MUST be NULL.

Event Object Type = 0x00000004 (Site)

MUST be NULL.

Event Object Type = 0x00000008 (Site Collection)

MUST be NULL.

Event Object Type = 0x00000010 (File)

MUST be NULL.

Event Object Type = 0x00000020 (Folder)

MUST be NULL.

Event Object Type = 0x00000040 (Alert Subscription)

MUST be NULL.

Event Object Type = 0x00000080 (Security Principal)

MUST be NULL.

Event Object Type = 0x00000100 (Security Group)

MUST be NULL.

Event Object Type = 0x00000200 (Content Type)

Event type	Description
0x00001000	The content type identifier of the content type.
0x00002000	The content type identifier of the content type.
0x00004000	The content type identifier of the content type.

Event Object Type = 0x00000400 (Field Template)

Event type	Description
0x00000400	The identifier of the field.

Event Object Type = 0x00000800 (Security Policy)

MUST be NULL.

Event Object Type = 0x00001000 (View)

MUST be NULL.

2.2.2.7 Change Log ItemFullUrl

Change log **ItemFullUrl** is a **Uniform Resource Locator (URL)** of **store-relative form**. It is part of a change log entry. This data has different meanings based on the event object type and event type data of the same change log entry. The possible meanings of this value are specified as the following:

Event Object Type = 0x00000001 (list item)

Event type	Description
0x00000010	URL in Store-relative form of the document
0x00000020	URL in Store-relative form of the document
0x00000040	URL in Store-relative form of the document
0x00000080	URL in Store-relative form of the document
0x00000100	URL in Store-relative form of the document
0x00001001	URL in Store-relative form of the document
0x00002000	MUST be NULL
0x00002002	URL in Store-relative form of the document
0x00004004	URL in Store-relative form of the document
0x00008000	The new URL in Store-relative form of the document
0x00010000	MUST be NULL
0x00020009	URL in Store-relative form of the document
0x00080000	URL of the security scope where the role assignment is added. This value MUST be NULL if the Change Log Int0 of the same Change Log Entry is also NULL
0x00100000	URL in Store-relative form of the document
0x02000000	URL in Store-relative Form to the security scope.
0x04000000	MUST be NULL

Event Object Type = 0x00000002 (List (1))

Event type	Description
0x00001000	MUST be NULL
0x00002000	MUST be NULL
0x00004000	URL in Store-relative form of the list (1)
0x00008000	MUST be NULL
0x00020000	MUST be NULL
0x00080000	URL of the security scope where the role assignment is added. This value MUST be NULL if Change Log Int0 of the same Change Log Entry is NULL
0x02000000	URL in Store-relative Form to the security scope.

Event Object Type = 0x00000004 (Site (2))

Event type	Description
0x00001000	MUST be NULL
0x00002000	MUST be NULL
0x00004000	URL in Store-relative form of the site (2), if this site (2) is deleted. This value MUST be NULL if the Site is converted into a folder under its parent site .
0x00008000	MUST be NULL
0x00040000	URL in the Store-relative Form of this site (2)
0x00020000	MUST be NULL
0x00800000	URL in Store-relative Form to the security scope.
0x01000000	URL in Store-relative Form to the security scope.
0x02000000	URL in Store-relative Form to the security scope.
0x00080000	URL in Store-relative Form to the security scope.
0x08000000	MUST be NULL

Event Object Type = 0x00000008 (Site collection)

MUST be NULL.

Event Object Type = 0x00000010 (File)

SHOULD be NON-NULL. If NOT NULL it MUST be store-relative form of the document.

Event Object Type = 0x00000020 (Folder)

Event type	Description
0x00001000	URL in Store-relative form of the Folder

Event type	Description
0x00002000	URL in Store-relative form of the Folder
0x00004000	URL in Store-relative form of the Folder
0x00008000	New URL in Store-relative form of the Folder
0x00010000	New URL in Store-relative form of the Folder after the move
0x00020000	URL in Store-relative form of the Folder
0x00100000	URL in Store-relative form of the Folder
0x04000000	URL in Store-relative form of the Folder after the move

Event Object Type = 0x00000040 (Alert subscription)

MUST be NULL.

Event Object Type = 0x00000080 (Security principal (2))

MUST be NULL.

Event Object Type = 0x00000100 (Security group)

MUST be NULL.

Event Object Type = 0x00000200 (Content type)

MUST be NULL.

Event Object Type = 0x00000400 (site column)

MUST be NULL.

Event Object Type = 0x00000800 (Security policy)

MUST be NULL.

Event Object Type = 0x00001000 (View)

MUST be NULL.

2.2.2.8 Change Log TimeLastModified

Change log **TimeLastModified** is a timestamp and is part of a change log entry. This data has different meanings based on the event object type and event type data of the same change log entry. The possible meanings of this value are specified as the following:

Event Object Type = 0x00000001 (List item)

Event type	Description
0x00000010	Time when the comment is created.
0x00000020	Time when the comment is updated.

Event type	Description
0x00000040	Time when the comment is deleted.
0x00000080	Time when the comment is closed.
0x00000100	Time when the comment is activated.
0x00001001	Time when the Add happened.
0x00002000	Time when the event (1) happened.
0x00002002	Time when the list item is updated.
0x00004004	Time when the delete happened.
0x00008000	Time when the rename happened.
0x00010000	Time when the restore happened.
0x00020009	Time when the event happened.
0x00080000	Time when this update happened.
0x00100000	Time when the update happened.
0x02000000	Time when the event (1) happened.
0x04000000	Time when the move happened.

Event Object Type = 0x00000002 (List (1))

Event type	Description
0x00001000	Time when the list (1) was created.
0x00002000	Time when the update happened.
0x00004000	Time when the list (1) was deleted.
0x00008000	Time when the list (1) was renamed.
0x00020000	Time of occurrence when the list was restored from the Recycle Bin .
0x00080000	Time when the event (1) happened.
0x02000000	Time when the event (1) happened.

Event Object Type = 0x00000004 (Site (2))

Event type	Description
0x00001000	Time when the site (2) was created.
0x00002000	Time of the update.
0x00004000	Time when the event (1) happened.
0x00008000	Time when the site (2) was renamed.

Event type	Description
0x00040000	Time when the event (1) happened.
0x00020000	Time when the event (1) happened.
0x00800000	Time when the event (1) happened.
0x01000000	Time when the event (1) happened.
0x02000000	Time when the event (1) happened.
0x00080000	Time when the event (1) happened.
0x08000000	Time when the site (2) navigation was updated.

Event Object Type = 0x00000008 (Site collection)

Event type	Description
0x00001000	Time when the creation happened
0x00002000	Time when the event (1) happened.
0x00004000	Time when the event (1) happened.
0x00008000	Time when the event (1) happened.

Event Object Type = 0x00000010 (File)

Event type	Description
0x00001000	Time when the addition happened.
0x00004000	Time when the deletion happened.
0x00010000	Time when the move happened.
0x00020000	Time when the restore occurred.
0x00008000	Time when the rename happens.
0x00100000	Time when the update happened.
0x04000000	Time when the move happened.

Event Object Type = 0x00000020 (Folder)

Event type	Description
0x00001000	Time when the restore happened.
0x00002000	Time when the update happened.
0x00004000	Time when the delete happened.
0x00008000	Time when the rename happened.

Event type	Description
0x00010000	Time when the folder was moved into a list (1).
0x00020000	Time when the rename happened.
0x00100000	Time when the folder is updated.
0x04000000	Time when the rename happened.

Event Object Type = 0x00000040 (Alert subscription)

Time when the event happened.

Event Object Type = 0x00000080 (Security principal (2))

Time when the event happened.

Event Object Type = 0x00000100 (Security group)

Time when the event happened.

Event Object Type = 0x00000200 (Content type)

Event type	Description
0x00001000	Time when the content type add happened.
0x00002000	Time when the content type update happened.
0x00004000	Time when the content type deletion happened.

Event Object Type = 0x00000400 (site column)

Time when the event happened.

Event Object Type = 0x00000800 (Security policy)

Time when the event happened.

Event Object Type = 0x00001000 (View)

Time when the event happened.

2.2.2.9 Change Log ItemName

Change log **ItemName** is a string and is part of a change log entry. This data has different meanings based on the event object type and event type data of the same change log entry. The possible meanings of this value are specified as the following:

Event Object Type = 0x00000001 (List item)

Event type	Description
0x00000010	Leaf name of the document.

Event type	Description
0x00000020	Leaf name of the document.
0x00000040	Leaf name of the document.
0x00000080	Leaf name of the document.
0x00000100	Leaf name of the document.
0x00001001	Leaf name of the document.
0x00002000	MUST be NULL.
0x00002002	Leaf name of the document.
0x00004004	Name of the object.
0x00008000	The Leaf Name of the List Item before it is renamed.
0x00010000	MUST be NULL.
0x00020009	List item name.
0x00080000	MUST be NULL.
0x00100000	List item name OR NULL.
0x02000000	MUST be NULL.
0x04000000	MUST be NULL.

Event Object Type = 0x00000002 (List (1))

MUST be NULL.

Event Object Type = 0x00000004 (Site (2))

Event type	Description
0x00001000	MUST be NULL.
0x00002000	MUST be NULL.
0x00004000	MUST be NULL.
0x00008000	MUST be NULL.
0x00040000	MUST be NULL.
0x00020000	MUST be NULL.
0x00800000	Role Name.
0x01000000	MUST be NULL.
0x02000000	MUST be NULL.

Event type	Description
0x00080000	MUST be NULL.
0x08000000	MUST be NULL.

Event Object Type = 0x00000008 (site collection)

MUST be NULL.

Event Object Type = 0x00000010 (File)

SHOULD be NULL.

Event Object Type = 0x00000020 (Folder)

MUST be NULL.

Event Object Type = 0x00000040 (Alert subscription)

MUST be NULL.

Event Object Type = 0x00000080 (Security principal (2))

MUST be NULL.

Event Object Type = 0x00000100 (Security group)

MUST be NULL except for the following event type:

Event type	Description
0x00004000	Group Title

Event Object Type = 0x00000200 (Content type)

MUST be NULL.

Event Object Type = 0x00000400 (site column)

MUST be NULL.

Event Object Type = 0x00000800 (Security policy)

MUST be NULL.

Event Object Type = 0x00001000 (View)

MUST be NULL.

2.2.2.10 Change Log Int1

Change log **Int1** is an integer and is part of a change log entry. This data has different meanings based on the event object type and event type data of the same change log entry. The possible meanings of this value are specified as the following:

Event Object Type = 0x00000001 (List item)

Event type	Description
0x00000010	MUST be NULL.
0x00000020	MUST be NULL.
0x00000040	MUST be NULL.
0x00000080	MUST be NULL.
0x00000100	MUST be NULL.
0x00001001	MUST be NULL.
0x00002000	MUST be NULL.
0x00002002	MUST be NULL.
0x00004004	MUST be NULL.
0x00008000	MUST be NULL.
0x00010000	MUST be NULL.
0x00020009	MUST be NULL.
0x00080000	If the value is NULL, it indicates that the anonymous user permission is updated. If this value is greater than or equal to 0x40000000, it indicates that a role assignment is added. If the value is NOT NULL and is less than 0x40000000, it is a Security Change Type Flag, as specified in section 2.2.3.3 , and indicates that a role inheritance has changed.
0x00100000	MUST be NULL.
0x02000000	If this value is not NULL, and is greater than or equal to 0x40000000, it is the Role Identifier of the role assignment being deleted. If this value is not NULL, and is less than 0x40000000, it MUST be a Security Change Type Flag 0x00000001, as specified in section 2.2.3.3 , indicating a role inheritance change. If this value is NULL, it indicates a security principal has been removed from all roles on the security scope.
0x04000000	MUST be NULL.

Event Object Type = 0x00000002 (List (1))

Event type	Description
0x00001000	MUST be NULL.
0x00002000	MUST be NULL.
0x00004000	MUST be NULL.
0x00008000	MUST be NULL.
0x00020000	MUST be NULL.
0x00080000	If the value is NULL, it indicates that the anonymous user permission is updated. If this value is greater than or equal to 0x40000000, it indicates that a role assignment is added. If the value is not NULL, and is less than 0x40000000, it is a Security Change Type Flag, as specified in section 2.2.3.3 , and indicates that a role inheritance has changed.

Event type	Description
0x02000000	If this value is not NULL, and is greater than or equal to 0x40000000, it is the Role Identifier of the role assignment being deleted. If this value is not NULL, and is less than 0x40000000, it MUST be a Security Change Type Flag 0x00000001, as specified in section 2.2.3.3 , indicating a role inheritance change. If this value is NULL, it indicates a security principal has been removed from all roles on the security scope.

Event Object Type = 0x00000004 (Site (2))

Event type	Description
0x00001000	MUST be NULL.
0x00002000	MUST be NULL.
0x00004000	MUST be NULL.
0x00008000	MUST be NULL.
0x00040000	If this value is greater than or equal to 0x40000000, it specifies the Identifier of the Role which is added. Otherwise, this is a Security Change Type Flag, as specified in section 2.2.3.3 .
0x00020000	MUST be NULL.
0x00800000	Role Identifier.
0x01000000	Role Identifier.
0x02000000	If this value is not NULL, and is greater than or equal to 0x40000000, it is the Role Identifier of the role assignment being deleted. If this value is not NULL, and is less than 0x40000000, it MUST be a Security Change Type Flag 0x00000001, as specified in section 2.2.3.3 , indicating a role inheritance change. If this value is NULL, it indicates a security principal has been removed from all roles on the security scope.
0x00080000	If the value is NULL, it indicates that the anonymous user permission is updated. If this value is greater than or equal to 0x40000000, it indicates that a role assignment is added. If the value is not NULL, and is less than 0x40000000, it is a Security Change Type Flag, as specified in section 2.2.3.3 , and indicates that a role inheritance has changed.
0x08000000	MUST be NULL.

Event Object Type = 0x00000008 (Site collection)

MUST be NULL.

Event Object Type = 0x00000010 (File)

MUST be NULL.

Event Object Type = 0x00000020 (Folder)

MUST be NULL.

Event Object Type = 0x00000040 (Alert Subscription)

MUST be NULL.

Event Object Type = 0x00000080 (Security Principal)

Event type	Description
0x00001000	MUST be NULL.
0x00002000	MUST be NULL, zero, or 1. If the value is 1, it indicates that the security principal's status as a site collection administrator has changed. If the value is zero or NULL, it has not changed.
0x00004000	MUST be NULL.

Event Object Type = 0x00000100 (Security Group)

MUST be NULL.

Event Object Type = 0x00000200 (Content Type)

MUST be NULL.

Event Object Type = 0x00000400 (Field Template)

MUST be NULL.

Event Object Type = 0x00000800 (Security Policy)

MUST be NULL.

Event Object Type = 0x00001000 (View)

MUST be NULL.

2.2.2.11 Change Log SiteId

Change log **SiteId** is a GUID and is part of a change log entry. This data has different meanings based on the event object type and event type data of the same change log entry. This value is the **site collection identifier** of the site collection under which the event (1) has occurred, except in the following cases, it is an **empty GUID**.

Event Object Type = 0x00000800 (Security Policy)

Event type	Description
0x00002000	MUST be an empty GUID.

Event Object Type = 0x00000008 (Site collection)

Event type	Description
0x00020000	When this value is not Empty GUID, the site collection identifier is restored. If this value is empty GUID, the content database containing this Change Log was restored.

2.2.2.12 Change Log WebId

Change log **WebId** is a GUID and is part of a change log entry. This data has different meanings based on the event object type and event type data of the same change log entry. This value is the **site identifier** of the site on which the event has occurred, except in the following cases, it MUST be NULL.

Event object type	Description
0x00000008	Site collection.
0x00000080	Security principal.
0x00000100	Security group.
0x00000800	Security policy.

2.2.3 Bit Fields and Flag Structures

2.2.3.1 Event Object Type Flags

A 4-byte unsigned integer bit mask that specifies the type of object upon which an event (1) has happened. The only valid values of the **event object type flags** bits are specified in the following table.

Value	Meaning
0x00000001	The event (1) is associated with a list item.
0x00000002	The event (1) is associated with a list (1).
0x00000004	The event (1) is associated with a site (2).
0x00000008	The event (1) is associated with a site collection.
0x00000010	The event (1) is associated with a file.
0x00000020	The event (1) is associated with a folder.
0x00000040	The event (1) is associated with an alert (1) .
0x00000080	The event (1) is associated with a user .
0x00000100	The event (1) is associated with a group (2) .
0x00000200	The event (1) is associated with a content type.
0x00000400	The event (1) is associated with a field (1) .
0x00000800	The event (1) is associated with a security policy.
0x00001000	The event (1) is associated with a view.

2.2.3.2 Event Type Flags

A 4 byte unsigned integer bit mask that specifies the type of an event (1) which can have one or more flags set. The only valid values of the **event type flags** bits are specified in the following table.

Value	Meaning
0x00000001	A list item is added.
0x00000002	A list item is modified.
0x00000004	A list item is deleted.
0x00000008	A list item is restored from a backup.
0x00000010	A discussion list item is added.
0x00000020	A discussion list item is modified.
0x00000040	A discussion list item is deleted.
0x00000080	A discussion list item is closed.
0x00000100	A discussion list item is activated.
0x00001000	A generic add event.
0x00002000	A generic modification event.
0x00004000	A generic delete event.
0x00008000	A generic rename event.
0x00010000	Move into.
0x00020000	Restore.
0x00040000	A permission level is added.
0x00080000	A role assignment is added.
0x00100000	A modification executed by the system.
0x00200000	A member (2) is added to a group (2).
0x00400000	A member (2) is deleted from a group (2).
0x00800000	A permission level is deleted.
0x01000000	A permission level is updated.
0x02000000	A role assignment is deleted.
0x04000000	Move away.
0x08000000	A navigation structure is changed.

2.2.3.3 Security Change Type Flags

A 32 bit mask that specifies modifications made to security settings. This flag **MUST** be used in conjunction with an event type flag. The valid values of the security change type flag bits **MUST** be one of the values in the following table.

Value	Meaning
0x00000000	No additional operation.
0x00000001	Remove role assignments on the current security scope and make it inherit role assignments from the parent security scope. This flag is meaningful only if the event type flag is 0x02000000.
0x00000002	The current site should define its own roles, instead of inheriting them from the parent site. This flag is meaningful only if the event type flag is 0x00040000.
0x00000004	The current security scope should define its own role assignments, instead of inheriting them from the parent security scope. This flag is meaningful only if the event type flag is 0x00080000.
0x00000008	Copy the roles defined on the parent site to this site. This flag is meaningful only if the event type flag is 0x00040000.
0x00000010	Copy the role assignments defined on the parent security scope to this security scope. This flag is meaningful only if the event type flag is 0x00040000 or 0x00080000.

2.2.3.4 Delete Flags

A 4-byte unsigned integer bit mask that specifies whether orphaned data is to be deleted for a site (2). In rare circumstances, while a site (2) is being deleted, the deletion operation could be preempted or failed in midstream. As a result, data for the Site could remain in the database. This remaining data is often referred to as orphaned data because it does not have context without the existence of its site. The only valid values of the **Delete Flags** bits are specified, as follows:

Value	Meaning
0	Do not delete orphaned data if the site's deletion operation was preempted or failed midstream.
8	Delete orphaned data if the site's deletion operation was preempted or failed midstream.

2.2.3.5 Document Flags

A 4-byte unsigned integer bit mask providing metadata about the document. This can have one or more flags set. The only valid values of the **document flags** bits are specified as follows:

Value	Description
0x00000001	This document contains dynamic content that SHOULD be sent through the CAML interpreter, an implementation-specific dynamic content generation component. An example of this would be a Category Web bot present in the source of the page.
0x00000002	The document is a "sub image" of another document. This is strongly correlated to the ExcludedType value in the security enumeration, and is set if this is an automatically generated thumbnail or Web image based on another item in the store.

Value	Description
0x00000004	The document is a type for which there was a registered parser available at the time it was saved. A parser is an implementation-specific component that can extract data and metadata from a document, which can then be used to build a list of hyperlinks and fields for content types.
0x00000008	The document is a type which can contain hyperlinks.
0x00000010	The document has an associated resource in the "_private" folder that should be renamed in parallel when this file is renamed. An example of this is the count file for a hit counter Web bot .
0x00000020	The document is currently checked out to a user.
0x00000040	The document is customized (1) .
0x00000080	The page contains Web Parts . Defaults to a personal view (showing Web Parts that are specific to the user that browsed to the page).
0x00000100	The document is a type which can have a binary stream.
0x00000200	The document is currently checked out to a location on the user's client system.
0x00000400	The document has child documents created by the document transformations feature.
0x00000800	The document is only a namespace entry for a list item. (in other words it corresponds to a list item in a list (1) that should be filtered out from file system-centric enumerations).
0x00001000	Unused.
0x00002000	The document has properties in its metadata defining a custom order of the content types. This is valid only for folders.
0x00004000	The document SHOULD be customized (1) when marked "not dirty" (in other words, when dependency updates are performed for the document). This is used for documents such as a document library template, which is provisioned as uncustomized but SHOULD be customized (1) to demote content type information about the containing document library whenever that information is updated.
0x00008000	Used when a zero-byte document is saved to a Document Library with required check-out and at least one required Field. This is common in migration scenarios or with the use of older versions of the WebDAV redirector against the wss Service WebDAV implementation.
0x00010000	DF_EXTSTORE
0x00020000	DF_SHARED_LOCK
0x00040000	The document is a Welcome page for the site (2) that contains it.
0x00080000	DF_ISPRIVATELISTEXEMPTFILE
0xFFFF0000	Currently unused and SHOULD be ignored.

2.2.4 Enumerations

2.2.4.1 Relationship Delete Behavior Type

It is an 8-bit integer enumeration of possible types of **relationship delete behaviors** for a **relationship lookup field**. The only valid values of the type of the relationship delete behavior are as follows.

Value	Meaning
1	The behavior is of type cascading behavior.
2	The behavior is of type restrict behavior.

2.2.5 Binary Structures

No common binary structures are defined in this protocol.

2.2.6 Result Sets

2.2.6.1 Site Collection Flags Result Set

The **Site Collection Flags Result Set** returns information about the site collection that has been deleted. The **Site Collection Flags Result Set** MUST return one row if the site collection specified by the **@SiteId** parameter existed in the content database and was deleted. If the site collection specified by the **@SiteId** parameter did not exist in the content database, the **Site Collection Flags Result Set** MUST NOT return any rows. The T-SQL syntax for the result set is as follows:

```
BitFlags int;
```

BitFlags: Contains the site collection flags of the site collection specified by the **@SiteId** parameter. For more information regarding site collection flags, see [\[MS-WSSFO2\]](#) section 2.2.2.9.

2.2.6.2 Distribution List E-mail Address Result Set

The **Distribution List E-mail Address Result Set** returns information about the site collection that has been deleted. If the site collection specified by the **@SiteId** parameter existed in the content database and was deleted, then the **Distribution List E-mail Address** result set MUST return one row for each of the site collection's security groups that contains a **distribution list e-mail address**. If, however, the deleted site collection's security groups do not contain any distribution list e-mail addresses, then the **Distribution List E-mail Address** result set MUST NOT return any rows. If the site collection specified by the **@SiteId** parameter did not exist in the content database then the **Distribution List E-mail Address** result set MUST NOT return any rows. The T-SQL syntax for the result set is as follows:

```
DLAlias nvarchar(128);
```

DLAlias: Contains a distribution list e-mail address for the deleted site collection specified by the **@SiteId** parameter.

2.2.6.3 Document Content Stream Transfer Result Set

The **Document Content Stream Transfer Result Set** returns information about the **document streams** in a site collection. Each row corresponds to a document stream along with associated metadata. The T-SQL syntax for the result set is as follows:

```
Size                int,
DocFlags            int,
{Content}           varbinary(max),
{RbsResReference}   varbinary(800),
Id                  uniqueidentifier,
InternalVersion     int,
Level               tinyint,
```

Size: The size, in bytes, of the document stream of the document.

DocFlags: The [Document Flags](#) for the document.

{Content}: The document's content stream. For an uncustomized document or if the document has external storage, this MUST be NULL. Otherwise, if the content is larger than the value specified in the *@ChunkSize* parameter, only the first *@ChunkSize* bytes MUST be returned, and the front-end Web server can request individual chunks of content in a subsequent request.

{RbsResReference}: If remote **binary large object (BLOB)** storage is enabled and the content of the document is contained in a remote data store, this MUST be the remote BLOB storage identifier for the document's content. If remote BLOB storage is disabled or the content of the document is not contained in a remote data store, this MUST be NULL. Further information about remote BLOB storage can be found in [\[MS-WSSO\]](#) section 2.1.2.3.8.

Id: The document identifier of the document.

InternalVersion: The internal version number of the document.

Level: The publishing level of the specified document.

2.2.6.4 Document Version Content Stream Transfer Result Set

The **Document Version Content Stream Transfer Result Set** returns information about the document streams for **document versions** in a site collection. Each row corresponds to a document stream for a document version along with associated metadata. The T-SQL syntax for the result set is as follows:

```
Size                int,
DocFlags            int,
{Content}           varbinary(max),
{RbsResReference}   varbinary(800),
Id                  uniqueidentifier,
InternalVersion     int,
UIVersion           int,
```

Size: The size, in bytes, of the document stream of the document version.

DocFlags: The [Document Flags](#) for the document version.

{Content}: The content stream of the document version. For an uncustomized document or if the document has external storage, this MUST be NULL. Otherwise, if the content is larger than the value specified in the *@ChunkSize* parameter, only the first *@ChunkSize* bytes MUST be returned, and the front-end Web server can request individual chunks of content in a subsequent request.

{RbsResReference}: If remote BLOB storage is enabled and the content of the document is contained in a remote data store, this MUST be the remote BLOB storage identifier for the document's content. If remote BLOB storage is disabled or the content of the document is not contained in a remote data store, this MUST be NULL. Further information about remote BLOB storage can be found in [\[MS-WSSO\]](#) section 2.1.2.3.8.

Id: The document identifier (2) of the document version.

InternalVersion: The **internal version number** of the document version.

UIVersion: The **user interface (UI) version** number of the specified document version.

2.2.6.5 Document Stream Storage Migration Result Set

The **Document Stream Storage Migration Result Set** returns information about the document streams in a site collection for remote data store management. Each row corresponds to a document stream along with associated metadata. The T-SQL syntax for the result set is as follows:

```
Size                int,  
DocFlags            int,  
{Content}           varbinary(max),  
{RbsResReference}  varbinary(800),  
Id                  uniqueidentifier,  
InternalVersion     int,  
Level               tinyint,  
SiteId              uniqueidentifier,  
DeleteTransactionId uniqueidentifier,  
ParentId            uniqueidentifier
```

Size: The size, in bytes, of the document stream of the document.

DocFlags: The [Document Flags](#) for the document.

{Content}: The document's content stream. For an uncustomized document or if the document has external storage, this MUST be NULL. Otherwise, if the content is larger than the value specified in the *@ChunkSize* parameter, only the first *@ChunkSize* bytes MUST be returned, and the front-end Web server can request individual chunks of content in a subsequent request.

{RbsResReference}: If remote BLOB storage is enabled and the content of the document is contained in a remote data store, this MUST be the remote BLOB storage identifier for the document's content. If remote BLOB storage is disabled or the content of the document is not contained in a remote data store, this MUST be NULL. Further information about remote BLOB storage can be found in [\[MS-WSSO\]](#) section 2.1.2.3.8.

Id: The document identifier (2) of the document.

InternalVersion: The internal version number of the document.

Level: The publishing level of the document.

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

DeleteTransactionId: The **delete transaction identifier** of the document.

ParentId: The document identifier (2) of the parent container of the document.

2.2.6.6 Document Version Stream Storage Migration Result Set

The **Document Version Stream Storage Migration Result Set** returns information about the document streams for document versions in a site collection for remote data store management. Each row corresponds to a document stream for a document version along with associated metadata. The T-SQL syntax for the result set is as follows:

```
Size                int,  
DocFlags            int,  
{Content}          varbinary(max),  
{RbsResReference} varbinary(800),  
Id                  uniqueidentifier,  
InternalVersion    int,  
UIVersion           int,  
SiteId              uniqueidentifier
```

Size: The size, in bytes, of the document stream of the document version.

DocFlags: The [Document Flags](#) for the document version.

{Content}: The content stream of the document version. For an uncustomized document or if the document has external storage, this MUST be NULL. Otherwise, if the content is larger than the value specified in the *@ChunkSize* parameter, only the first *@ChunkSize* bytes MUST be returned, and the front-end Web server can request individual chunks of content in a subsequent request.

{RbsResReference}: If remote BLOB storage is enabled and the content of the document is contained in a remote data store, this MUST be the remote BLOB storage identifier for the document's content. If remote BLOB storage is disabled or the content of the document is not contained in a remote data store, this MUST be NULL. Further information about remote BLOB storage can be found in [\[MS-WSSO\]](#) section 2.1.2.3.8.

Id: The document identifier (2) of the document version.

InternalVersion: The internal version number of the document version.

UIVersion: The UI version number of the document version.

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

2.2.6.7 File Fragment Result Set

The **File Fragment Result Set** contains information about the file fragments being fetched. If **@GetAfter** is 1, **@TopRows** number of file fragments with File Fragment Identifier greater than **@Id** MUST be fetched. Otherwise, if **@GetOnly** is 1, only one file fragment with File Fragment Identifier equal to **@Id** MUST be fetched. If both **@GetAfter** and **@GetOnly** are zero, **@Id** MUST be ignored and **@TopRows** number of file fragments MUST be fetched, starting from the smallest available File Fragment Identifier for the document specified by **@DocId**.

There MUST be one row for each file fragment being fetched. The result set MUST be sorted by File Fragment Identifier in ascending order.

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

```
Id                bigint,
Partition         tinyint,
Tag               varbinary(40),
BlobSize         int
{BlobData}       varbinary(max)
```

Id: The File Fragment Identifier of the file fragment.

Partition: The identifier of a file fragment partition for the **file fragment partition** in which the file fragment belongs.

Tag: The **file fragment tag** for the file fragment.

{BlobData}: The data for this file fragment.

BlobSize: The size in bytes of the entire data of this file fragment.

2.2.7 Tables and Views

2.2.7.1 AllUserData Table

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

2.2.7.2 NameValuePair Table

The **NameValuePair** table stores list item data for **indexed fields**. The **NameValuePair** table is defined using **Transact-Structured Query Language (T-SQL)** syntax, as follows:

```
TABLE GroupMembership(
  SiteId          uniqueidentifier NOT NULL,
  WebId           uniqueidentifier NOT NULL,
  ListId          uniqueidentifier NOT NULL,
  ItemId          int NOT NULL,
  Level           tinyint DEFAULT 1 NOT NULL,
  FieldId         uniqueidentifier NOT NULL,
  Value           sql_variant
);
```

SiteId: The site collection identifier of the site collection containing the list item.

WebId: The site identifier of the site containing the list item.

ListId: The **list identifier** of the list containing the list item.

ItemId: The item identifier of the list item.

Level: The publishing level of the list item.

FieldId: The **field identifier** of an indexed field of the list item.

Value: The value of the indexed field specified by the **FieldId** column.

2.2.7.3 NameValuePair_Latin1_General_CI_AS Table

The **NameValuePair_Latin1_General_CI_AS** table stores textual list item data for indexed fields using the **Latin1_General_CI_AS collation order**. The **NameValuePair_Latin1_General_CI_AS** table is defined using T-SQL syntax, as follows:

```
TABLE GroupMembership(  
    SiteId          uniqueidentifier NOT NULL,  
    WebId           uniqueidentifier NOT NULL,  
    ListId         uniqueidentifier NOT NULL,  
    ItemId         int NOT NULL,  
    Level          tinyint DEFAULT 1 NOT NULL,  
    FieldId       uniqueidentifier NOT NULL,  
    Value         nvarchar(255) COLLATE Latin1_General_CI_AS  
);
```

SiteId: The site collection identifier of the site collection containing the list item.

WebId: The site identifier of the site containing the list item.

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

ItemId: The item identifier of the list item.

Level: The publishing level of the list item.

FieldId: The field identifier of an indexed field of the list item.

Value: The value of the indexed field specified by the **FieldId** column.

2.2.7.4 Collated NameValuePair Tables

A table exists for each collation order specified in [\[MS-WSSFO2\]](#) section 2.2.3.4. These tables are identical to the **NameValuePair_Latin1_General_CI_AS** table, except that every reference to **Latin1_General_CI_AS** is replaced with the appropriate collation order name. For example, the **NameValuePair_Albanian_CI_AS** table uses the **Albanian_Ci_AS** collation order for the **Value** column.

2.2.8 XML Structures

2.2.8.1 Namespaces

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

2.2.8.2 Simple Types

This specification does not define any common XML Schema simple type definitions.

2.2.8.3 Complex Types

2.2.8.3.1 List Data Source

The following **XML schema definition (XSD)** defines the List Data Source Definitions:

```
<xs:element name="DataSource" type="ListDataSourceDefinitions" minOccurs="0" maxOccurs="1" />
<xs:complexType name="ListDataSourceDefinitions">
  <xs:sequence>
    <xs:element name="Property" type="ListDataSourcePropertyDefinition" minOccurs="0"
maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ListDataSourcePropertyDefinition">
  <xs:attribute name="Name" type="xs:string" />
  <xs:attribute name="Value" type="xs:string" />
</xs:complexType>
```

DataSource: The top element that represents the **data source** for the list (1).

Property: Represents one property of the data source.

Property.Name: The property's name. The **Name** attribute MUST have values in the set: "LobSystemInstance", "EntityNamespace", "Entity", and "SpecificFinder".

Property.Value: The property value.

Example:

```
<Properties>
  <Property Name="LobSystemInstance" Value="Talent Management - Seattle" />
  <Property Name="EntityNamespace" Value="TalentManagement" />
  <Property Name="Entity" Value="Agent" />
  <Property Name="SpecificFinder" Value="AgentSpecificFinder" />
</Properties>
```

2.2.8.4 Elements

This specification does not define any common XML Schema element definitions.

2.2.8.5 Attributes

This specification does not define any common XML Schema attribute definitions.

2.2.8.6 Groups

This specification does not define any common XML Schema group definitions.

2.2.8.7 Attribute Groups

This specification does not define any common XML Schema attribute group definitions.

3 Protocol Details

3.1 Server Details

3.1.1 Abstract Data Model

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

3.1.1.1 Change Log Operations

The protocol server stores a hierarchy of objects. The protocol server also maintains a Change Log table that records various Events that happened on those objects. The Events can be added directly by a protocol client using this protocol, or they can also be added indirectly when the client communicates with the server using a different protocol. For example, the client calls **proc_AddDocument** from [\[MS-WSSFO2\]](#) section 3.1.5.3 which will result in an Event being added to the change. In addition, the client can use this protocol to retrieve information about Events that are currently in the Change Log. This documents specifies different types of events and how the information returned by the back-end database server should be interpreted.

3.1.1.2 Publish and Un-publish Operations

The back-end database server stores a collection of documents. Each document can exist in up to three different publishing levels: Checked Out, **draft**, and **published**, each with their own copy of the document and associated information. A document in the Draft publishing level typically has restricted visibility compared to a document in the Published publishing level. As part of this protocol, a front-end Web server can publish a document to change the document's **current version** from draft to published, or undo a publish to take the current version from Published to Draft. The following diagram illustrates this process.

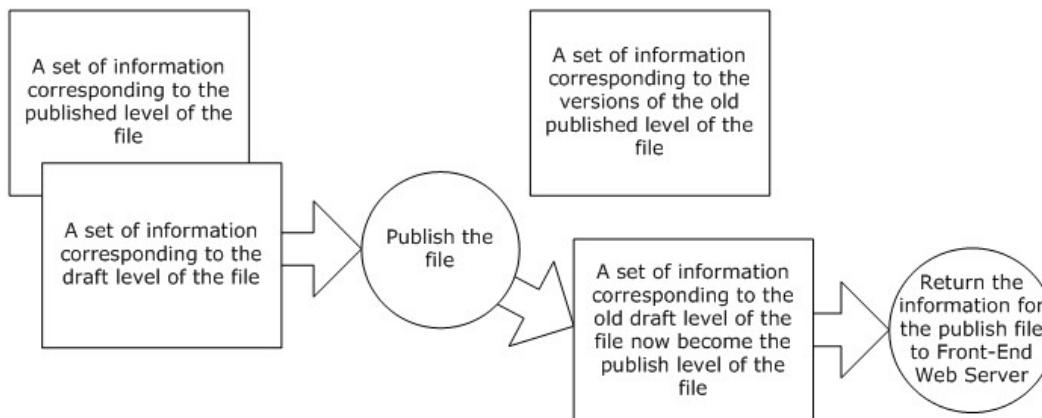


Figure 2: Publish and un-publish operations

3.1.1.3 Check-In and Check-Out Operations

The front-end Web server can update the back-end database server to set a document's current version to "Checked Out", which creates a separate logical copy of that document and associated

information in the Checked Out publishing level. As part of this check-out operation, the back-end database server stores a **user identifier** for that document recording the user that has that document checked out. Only that user can view the copy of the document that is in the Checked Out publishing level. If the check-out operation is a short-term check-out, the back-end database server also stores a time at which the check-out expires. The following diagram illustrates this process.

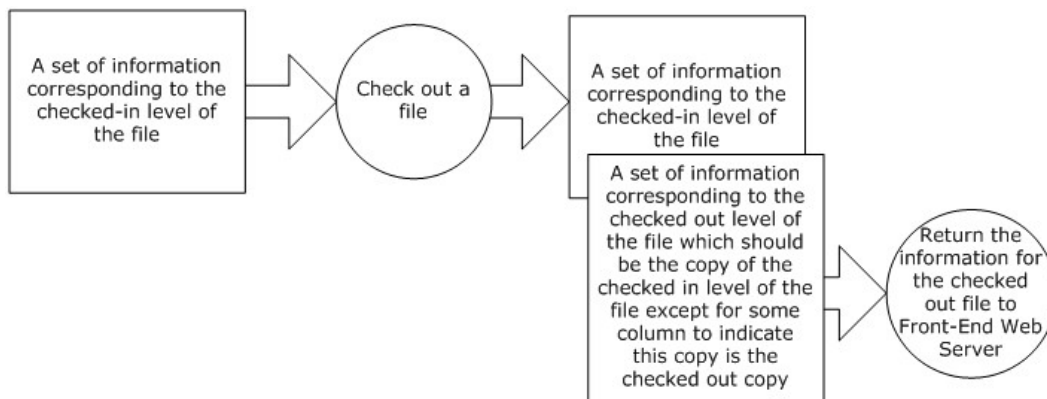


Figure 3: Check-out operations

The front-end Web server can update the back-end database server to check in a document that is currently in the Checked Out publishing level. The publishing level for the document is updated on the back-end database server to be either "Published" or "Draft", as requested.

3.1.1.4 Historical Versioning Operations

The back-end database server maintains a (possibly empty) collection of historical versions for each document, containing information associated with previous revisions to the document. As part of this protocol, the front-end Web server can enumerate the collection of historical versions stored on the back-end database server as illustrated in the following diagram.

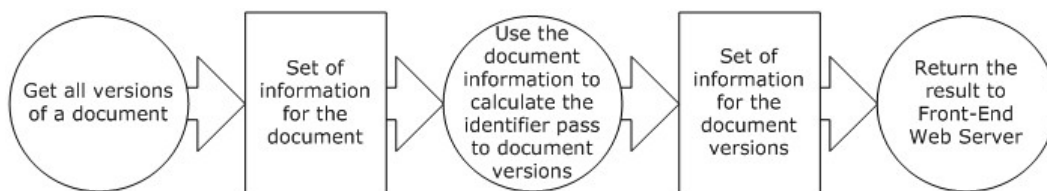


Figure 4: Historical versioning operations

The front-end Web server can also use this protocol to delete historical versions on the back-end database server or mark them as deleted in the Recycle Bin. As part of checking in a document, the front-end Web server can create a new historical version, copying the current version of the document and adding it to the collection of historical versions maintained by the back-end database server.

3.1.1.5 Link Fixup Operations

When a protocol client starts a link fixup operation, it starts the operation with a call to **proc_StartUndirtyList** call and retrieves the data columns from the [AllUserData](#) table that correspond to fields whose type allows forward links to be discovered for rows whose corresponding entry in the **AllDocs** table. Given the resulting range, the protocol client calls

proc_GetListDataLinks to obtain the correct forward links for the data. The protocol client then computes the correct values for the field data and commits the data back to the AllUserData table, also calling **proc_UndirtyListItem** for each item. After repeating this procedure for every dirty list item, the protocol client includes a call to **proc_FinishUndirtyList** to complete the operation.

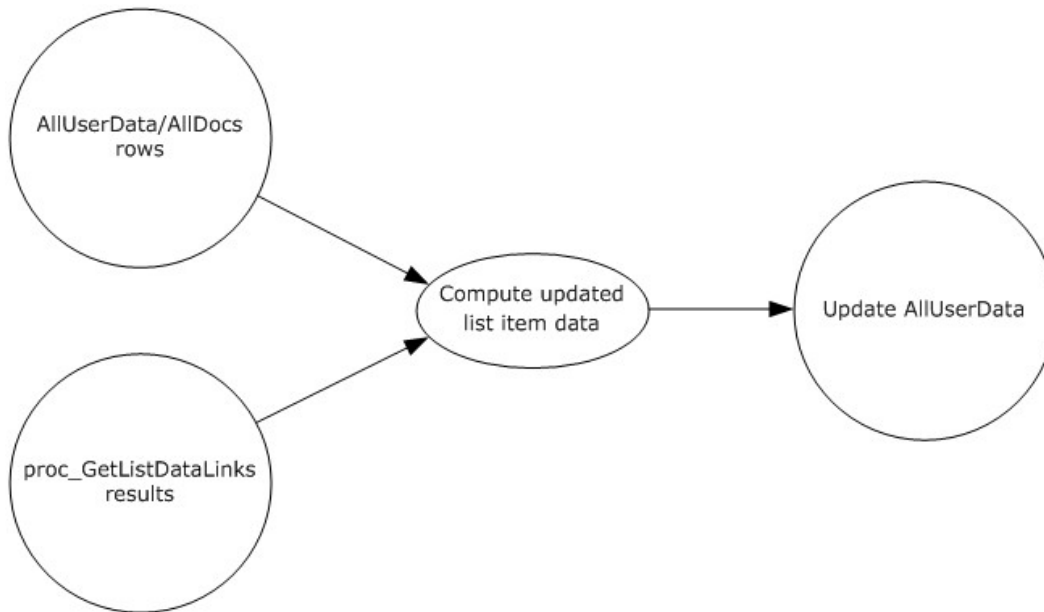


Figure 5: Link fixup operations

3.1.1.6 Lookup Relationships Operations

The back-end database server stores relationships between lists (1) using relationship lookup fields. A relationship lookup field defines a relationship between a list (1) and a target list (1) and can have either of two behaviors defined on it: cascading behavior and restrict behavior. The behavior defined on a relationship lookup field defines what **MUST** happen when a list item or document is deleted in the target list (1).

As part of this protocol, a front-end Web server can add, change or remove behaviors for relationship lookup fields. The front-end Web server can trigger a cascading delete operation by finding what list items or documents to delete as part of the protocol, and then deleting the list items or documents involved in a single batch. On the other hand, the front-end Web server can detect whether a list item or document has any list items or documents looking up to it and hence, restrict the deletion of the list item.

3.1.1.7 Theme Operations

Theme data for this protocol is maintained on both the front-end Web server and back-end database server. The front-end Web server stores the content of the theme files in its **file system**. The back-end database server stores additional theme metadata in one or more content databases. The theme metadata stored in the appropriate content databases is created and maintained when theme data is loaded and applied to a site.

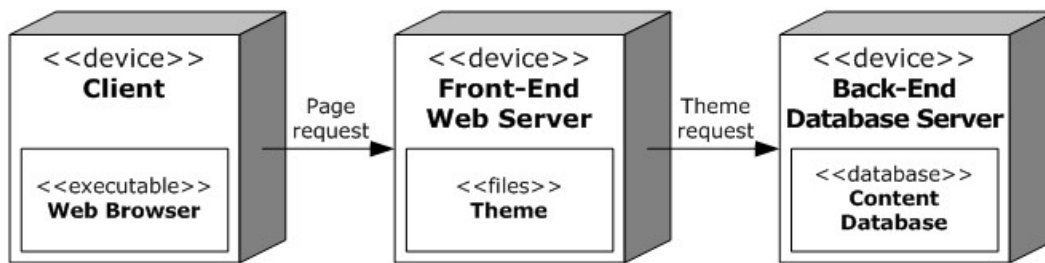


Figure 6: Theme operations

Client: The client referred to in the previous diagram is the computer that will be requesting a page for a site from the front-end Web server.

Web Browser: A software application capable of displaying HTML pages requested by the front-end Web server.

Theme: A collection of graphics and Cascading Style Sheets (CSS) that determine visual aspects of pages for a site. For example, a theme can determine the background color of a page, the page text color, font, and alignment, how **hyperlinks** on pages change in behavior when clicked on by a mouse, and the presence of tooltips when a mouse is hovered over text or images on pages.

Theme Files: Consists of images and CSS files that comprise a theme. These files are stored on the front-end Web server file system and their contents are retrieved when the client's Web browser when requesting a page from the front-end Web server.

Theme Name: A Unicode string that uniquely identifies the set of theme files for a theme.

Theme Metadata: The theme metadata is comprised of information about the theme files stored on the front-end Web server. The theme metadata is stored in the content database on the back-end database server. Theme metadata consists of the following:

- The URL in of the theme files in store-relative form.
- The theme name.
- The installation address in which theme files can be found on the file system of the front-end Web server.

Theme Installation Path: The directory path fragment in which the theme files can be found on the front-end Web server file system for a theme. For example, the theme files for the "Wheat" theme would found at Program Files\Common Files\ Shared\Web Server Extensions\12\template\themes\wheat\.

Theme XML File: The theme XML file is an **XML** file located on the file system of the front-end Web server. For example, if the theme XML file is translated into the 1033 **language code identifier (LCID)**, then the file would be located at Program Files\Common Files\Microsoft Shared\Web Server Extensions\12\layouts\1033\ sphemes.xml.

Theme XSD File: The theme XML file conforms to the **XML schema** defined by the theme XSD file. This file is located on the front-end Web server file system at the same location as that of the Theme XML File where LCID is the language code identifier (LCID) that the theme XSD file has been translated into. For example, if the theme XSD file is translated into the 1033 LCID, the file is located at Program Files\Common Files\Microsoft Shared\Web Server Extensions\12\layouts\1033\ sphemes.xsd.

3.1.1.8 Wide List Operations

Wide lists are created when the number of fields required by a list of a specific type exceeds a fixed maximum per type allowed per row in the content database.

The following table lists the number of fields allowed, per type of field, per row.

nvarchar(255)	64
Ntext	32
sql_variant	8
Int	16
Float	12
datetime	8
Bit	16
Uniqueidentifier	1

3.1.1.9 File Fragment Operations

The back-end database stores file fragments associated with an existing document. When a client sends a request to a front-end Web server, it can either update the content of the document or add, delete, or update file fragments instead. Other clients can request all or a specific group of file fragments for the document. When the client updates the content of the document, it may also specify which file fragments to delete in which file fragment partitions via arguments to **proc_UpdateDocument** in [\[MS-WSSFO2\]](#) section 3.1.5.128. This is because file fragments might only be associated with a specific state of a document, so once the contents change, the fragments might no longer be valid.

3.1.2 Timers

An execution timeout timer on the protocol server governs the execution time for the client's 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 columns 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 fn_RoundDateToNearestSecond

The **fn_RoundDateToNearestSecond** function is called to round a date and time value to the nearest second.

The T-SQL syntax for the function is as follows.

```
FUNCTION fn_RoundDateToNearestSecond(  
    @x                datetime  
)  
RETURNS              datetime;
```

@x: A date and time value.

Return value: The function MUST return the date and time value that is @x rounded to the nearest second.

3.1.4.2 fn_UnpackCsvString

The **fn_UnpackCsvString** function is called to parse a comma-separated string.

The T-SQL syntax for the function is as follows.

```
FUNCTION fn_UnpackCsvString(  
    @list            nvarchar(max)  
)  
RETURNS            TABLE (val nvarchar(255));
```

@list: A string of comma-separated values.

Return values: The function MUST return a table that contains all the substrings separated by commas in the input, each substring in a table row, and the order of table rows is identical to the order of substrings in **@list**.

3.1.4.3 proc_AddDependency

The **proc_AddDependency** stored procedure is called to create a dependency for a document on some other property so that the document is processed after the property is changed. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_AddDependency(  
    @SiteId            uniqueidentifier,  
    @FullUrl           nvarchar(260),  
    @Level             tinyint,  
    @DepType           tinyint,  
    @DepDesc           nvarchar(270), @RequestGuid    uniqueidentifier = NULL OUTPUT  
);
```

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

@FullUrl: The URL of the specified document in store-relative form for which the document dependency will be created.

@Level: The publishing level of the specified document.

@DepType: The dependency type. The following values are valid:

Value	Description
1	Document dependency. This updates items dependent on the specified document. The @DepDesc parameter is the store-relative form URL of the document that has changed.
3	Configuration dependency. This updates items dependent on changes to system configuration metadata, as specified in [MC-FPSEWM] , section 6. The @DepDesc parameter is the meta-key for the metadata that has changed.
4	Navigation dependency. This updates items dependent on changes to navigation structures. The @DepDesc parameter contains the Web-Navigation-URL, as specified in [MC-FPSEWM] section 2.2.2.2.34, for a navigation structure node.
7	Usage dependency. This updates items dependent on changes to site usage statistics. The @DepDesc parameter is the store-relative form URL of the site.

@DepDesc: The dependency description, which varies according to the value of **@DepType**, as described in the preceding table.

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

Return values: An integer that the protocol client MUST ignore.

Result sets: MUST NOT return any result set.

3.1.4.4 **proc_AddEventToCache**

The **proc_AddEventToCache** stored procedure is called to add an event (1) to the back-end database server. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_AddEventToCache (
    @SiteId          uniqueidentifier,
    @WebId           uniqueidentifier,
    @ListId          uniqueidentifier,
    @ItemId          int,
    @ItemName        nvarchar(255),
    @ItemFullUrl     nvarchar(260),
    @DocId           uniqueidentifier,
    @EventType       int,
    @ModifiedBy      nvarchar(255),
    @TimeLastModified datetime,
    @EventData       varbinary(max),
    @ACL             varbinary(max),
    @ScopeId        uniqueidentifier = NULL,
    @DocClientId     varbinary(16) = NULL,
    @RequestGuid     uniqueidentifier = NULL OUTPUT
);
```

@SiteId: The site collection identifier of the site collection in which the specified event (1) has occurred.

@WebId: The site identifier of the site (2) in which the specified event (1) has occurred.

@ListId: The list identifier of the list (1) to which the event (1) is related.

@ItemId: The **list item identifier** from the list (1) specified by **@ListId** that is associated with the event (1). Its value **MUST** be -1 or a list item identifier.

@ItemName: A string that represents the name of the list item for which the event (1) has occurred, or NULL if the event (1) did not pertain to a list item.

@ItemFullUrl: The Uniform Resource Locator (URL) associated with the event (1) in store-relative form. This parameter **MUST** be NULL if **@ItemId** is not -1. When **@ItemId** is -1, this parameter **MUST NOT** be NULL.

@DocId: This parameter **MUST** be NULL. .

@EventType: An integer that specifies [Event Type Flags](#).

@ModifiedBy: A string which specifies the **login name** of a security principal (2) who added this event (1).

@TimeLastModified: A timestamp in **Coordinated Universal Time (UTC)** that specifies the time when this event (1) occurred.

@EventData: This parameter contains implementation-specific event (1) data.

@ACL: A byte array in the **access control list (ACL)** format. If this parameter is NULL, this stored procedure will use the ACL defined on the security scope specified by the **@ScopeId** parameter. If the **@ScopeId** parameter is also NULL, the stored procedure will use the **ACL** from the object specified by **@ItemFullUrl**.

@ScopeId: This parameter **MUST** be NULL.

@DocClientId: This parameter **MUST** be NULL.

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

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

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

3.1.4.5 proc_AddGhostDocument

The **proc_AddGhostDocument** stored procedure is called to create an uncustomized file. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_AddGhostDocument (
    @SiteId                uniqueidentifier,
    @WebId                 uniqueidentifier,
    @DocId                 uniqueidentifier,
    @DocDirName            nvarchar(256),
    @DocLeafName           nvarchar(128),
    @Level                 tinyint,
    @UIVersion             int,
    @EnableMinorVersions   bit,
    @DocSize               int,
    @DocFlags              int,
    @OnRestore             bit = 0,
```

```

@Overwrite          bit OUTPUT,
@UserId            int,
@HasDeleteListItemsRight bit,
@UpdateSiteQuota  bit,
@SetupPathVersion tinyint,
@SetupPath         nvarchar(255),
@SetupPathUser    nvarchar(255),
@ListId           uniqueidentifier = NULL,
@DoclibRowId      int = NULL,
@fCheckQuotaAndWriteLock bit = 0,
@fCheckIfWebWelcomePage bit = 0,
@DTM              datetime = NULL OUTPUT
@RequestGuid      uniqueidentifier = NULL OUTPUT
);

```

@SiteId: The site collection identifier of the site collection that contains the site (2).

@WebId: The site identifier of the site (2) which will contain the uncustomized file.

@DocId: The document identifier (2) for the new uncustomized file to be created. An existing file with the specified document identifier (2) MUST NOT exist.

@DocDirName: The **directory name** for the new uncustomized file.

@DocLeafName: The leaf name for the new uncustomized file.

@Level: The publishing level for the new uncustomized file. Refer to [\[MS-WSSFO2\]](#) section 2.2.2.6.

@UIVersion: The user interface (UI) version number for the new uncustomized file.

@EnableMinorVersions: This parameter is not used and MUST be ignored.

@DocSize: The size, in bytes, of the new uncustomized file.

@DocFlags: The document flags, as specified in section [2.2.3.5](#), for the new uncustomized file to be created.

@OnRestore: Whenever an uncustomized file is successfully created, an event object type flag, as specified in section [2.2.3.1](#), of "0x00000010" with an event type flag, as specified in section [2.2.3.2](#), of "0x00001000" is recorded in the change log with the **datetime** in Coordinated Universal Time (UTC) when the uncustomized file was created. An additional event object type flag of "0x00000010" with an event type flag of "0x00100000" is recorded in the change log. If this parameter is "1", the datetime recorded is NULL. No datetime is recorded in the change log. If this parameter is set to something other than "1", the datetime recorded is the current **UTC**.

@Overwrite: If this parameter is set to a value other than "1" on input, this parameter MUST be ignored. However, if it is set to "1" on input and an existing file is found specified by the **@DocDirName** and **@DocLeafName** parameters and the **@HasDeleteListItemsRight** parameter is set to "1", the existing file is deleted and replaced with the new uncustomized file being created. If the existing file is successfully deleted, the **@Overwrite** parameter is set to "1" on output. If the existing file is not successfully deleted, the **@Overwrite** parameter is set to zero ("0") on output.

@UserId: The user identifier of the user requesting the operation. If the specified **@Level** parameter is set to publishing level of type draft, then the draft owner for the file will be set to this user identifier. If the specified **@Level** parameter is set to publishing level of checked out, then the checked out owner for the file will be set to this user identifier.

@HasDeleteListItemsRight: This parameter is only used if the *@Overwrite* parameter is set to "1" on input. When set to "1", the impact of this parameter is described in the definition for the *@Overwrite* parameter.

@SetupPathVersion: This parameter specifies the directory path fragment where the uncustomized file content is located on the front-end Web server's file system. This parameter MUST be one of the values in the following table.

Value	Description
"2"	The <i>@SetupPath</i> parameter value supplied is relative to the install location of Windows® SharePoint® Services 2.0 on the front-end Web server (for example Program Files\Common Files\Microsoft Shared\Web Server Extensions\60).
"3"	The @SetupPath parameter value supplied is relative to the install location of Windows® SharePoint® Services 3.0 on the front-end Web server (for example Program Files\Common Files\Microsoft Shared\Web Server Extensions\12).
"4"	The @SetupPath parameter value supplied is relative to the install location of Microsoft® SharePoint® Foundation 2010 on the front-end Web server (for example Program Files\Common Files\Microsoft Shared\Web Server Extensions\14).

@SetupPath: This specifies the **Unicode** directory path fragment relative to the base directory path specified by the *@SetupPathVersion* parameter. Taken together, the *@SetupPath* and *@SetupPathVersion* parameters specify where the uncustomized file can be found on the front-end Web server's file system.

@SetupPathUser: This specifies the Unicode of the login name of the user that is creating the uncustomized file.

@ListId: This parameter is optional and defaults to NULL if not specified. If this parameter is not NULL, it MUST be the list identifier of a list (1) contained within the specified site. If this parameter is NULL, the uncustomized file to be created will not be contained within a list (1).

@DoclibRowId: This parameter is optional and defaults to NULL if not specified. If this parameter is not NULL, then it MUST be the document library row identifier for the new uncustomized file. If the uncustomized file will not be contained within a list (1), this parameter and the **@ListId** parameter MUST be NULL.

@fCheckQuotaAndWriteLock: This parameter is optional and defaults to zero ("0") if not specified. If this parameter is set to zero ("0"), it MUST be ignored. However, if it is set to a value other than zero, the stored procedure verifies that adding the uncustomized file will not exceed the specified site collection's **site collection quota** and that the site collection's **site collection flag** does not contain the 0x00000001 bit. As specified in [MS-WSSFO2], section [2.2.2.9](#).

@fCheckIfWebWelcomePage: This parameter is optional and defaults to zero ("0") if not specified. If this parameter is set to zero, it MUST be ignored. However, if it is set to a value other than zero, the stored procedure compares the URL of the uncustomized file to the URL of the Welcome page of the site (2). If the Welcome page exists and its URL matches the URL of the uncustomized page, the document flag 0x00040000 is set to "1". If the Welcome page does not exist or the URL of Welcome page does not match the URL of the uncustomized page, the document flag 0x00040000 is set to zero ("0").

@DTM: This parameter is optional and defaults to NULL if not specified. If specified, the input value is ignored and the value set on output will be the datetime in UTC when the uncustomized file is created.

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

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

Value	Description
"0"	Successful execution.
"3"	This return code MUST be returned if any of the following are true: <ul style="list-style-type: none">▪ If the site collection specified by the <i>@SiteId</i> parameter does not exist.▪ If the site (2) specified by the <i>@WebId</i> parameter does not exist.▪ The site collection specified by the <i>@SiteId</i> parameter exists, but its directory name does not match the value specified by the <i>@DocDirName</i> parameter.
"5"	This return code is returned if the uncustomized file failed to be created. This is because either the user identifier specified by the <i>@UserId</i> parameter does not have the appropriate permissions, or the <i>@Overwrite</i> parameter is set to "1" and the <i>@HasDeleteListItemsRight</i> parameter is set to "1" and an existing file was not found specified by the <i>@DocDirName</i> and <i>@DocLeafName</i> parameters.
"80"	The uncustomized file failed to be created because an existing file was found specified by the <i>@DocDirName</i> and <i>@DocLeafName</i> parameters and the <i>@Overwrite</i> parameter was set to a value other than "1".
"212"	If the <i>@fCheckQuotaAndWriteLock</i> parameter is set to any value other than zero ("0"), the creation of the uncustomized file will fail with this return code if the specified site collection has its WRITELOCK (0x00000001) site collection flag bit set. As specified in [MS-WSSFO2], section 2.2.2.9 .
"1816"	If the <i>@fCheckQuotaAndWriteLock</i> parameter is set to any value other than zero ("0"), the creation of the uncustomized file will fail with this return code if adding the uncustomized file will exceed the site collection quota for the specified site collection.

Result sets: MUST NOT return any result sets.

3.1.4.6 **proc_AddNewRowOrdToList**

The **proc_AddNewRowOrdToList** stored procedure is called to allocate more storage per list item for a wide list. For an example, see section [4.4](#). The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_AddNewRowOrdToList (  
    @SiteId                uniqueidentifier,  
    @WebId                 uniqueidentifier,  
    @ListID               uniqueidentifier,  
    @RowOrdinal           int,  
    @CheckSchemaVersion   int NULL,  
    @RequestGuid          uniqueidentifier = NULL OUTPUT  
);
```

@SiteId: The site collection identifier of the site collection which contains the specified site (2).

@WebId: The site identifier of the site (2) which contains the specified list (1).

@ListID: The list identifier of the list (1) for which the new row ordinal is added.

@RowOrdinal: The next maximum row ordinal to use for this list (1). This parameter MUST be greater than one and MUST be at most one greater than the current maximum row ordinal for the list (1).

@CheckSchemaVersion: The version number of a list schema to verify against to ensure no intervening change to the **list schema** was made. If this parameter is not NULL, and **@CheckSchemaVersion** is older than the current version, then 1638 is returned.

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

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

Value	Description
0	Successful execution.
87	The operation could not be finished because of invalid parameters.
1638	The schema version of the list has changed, and the operation cannot continue.

Result sets: MUST NOT return any result sets.

3.1.4.7 **proc_AddNewRowOrdToListItem**

The **proc_AddNewRowOrdToListItem** stored procedure is called to add one row for a list item in a wide list. For an example, see section [4.5](#). The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_AddNewRowOrdToListItem(  
    @SiteId                uniqueidentifier,  
    @WebId                  uniqueidentifier,  
    @ListID                 uniqueidentifier,  
    @ItemID                 int,  
    @RowOrdinal             int,  
    @CheckSchemaVersion    int = NULL,  
    @RequestGuid            uniqueidentifier = NULL OUTPUT  
);
```

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

@WebId: The site identifier of the site (2) which contains the specified list (1).

@ListID: The list identifier of the list (1) that contains the list item for which the new row is added.

@ItemID: The list item identifier of the list.

@RowOrdinal: The row ordinal for the list item. This parameter MUST be greater than 1 and MUST be at most the current maximum row ordinal for the list (1).

@CheckSchemaVersion: The version number of a list schema to verify against to ensure no intervening change to the list schema was made.

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

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

Value	Description
0	Successful execution.
87	The operation could not be finished because of invalid parameters.
1638	The schema version of the list has changed, and the operation cannot continue.

Result sets: MUST NOT return any result sets.

3.1.4.8 proc_AL

The **proc_AL** stored procedure is called to store a link from a document, list (1), folder, or **document library** to another document, list (1), folder, or document library. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_AL(
    @SiteId           uniqueidentifier,
    @DirName          nvarchar(256),
    @LeafName         nvarchar(128),
    @Level            tinyint,
    @LinkNumber       int,
    @TargetDirName    nvarchar(256),
    @TargetLeafName   nvarchar(128),
    @Type             tinyint,
    @Security         tinyint,
    @Dynamic          tinyint,
    @ServerRel        bit,
    @Search           nvarchar(max),
    @WP               uniqueidentifier = NULL,
    @Fld              uniqueidentifier = NULL,
    @RequestGuid      uniqueidentifier = NULL OUTPUT
);

```

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

@DirName: The directory name of the location of the document that contains the link to be stored. This MUST NOT be NULL.

@LeafName: The leaf name of the document that contains the link. This MUST NOT be NULL.

@Level: The publishing level of the source object.

@LinkNumber: Ordinal number of the link in the document. This value MUST NOT be NULL.

@TargetDirName: The directory name of the linked document or list (1) or folder. This value MUST NOT be NULL.

@TargetLeafName: The leaf name of the linked document or list (1) or folder. This value MUST NOT be NULL.

@Type: A 1- byte (**tinyint**) value represented as a single upper case **ASCII** character specifying the link type. The value MUST be one of the values in the following table.

Value	Description
"A"	The link is from the ACTION attribute of an HTML FORM tag.
"B"	The link is from the attribute markup of a bot.
"C"	The link is from an auto-generated table of contents. Agents can ignore the link type when determining unreferenced files within a site (2).
"D"	The link references programmatic content, as in the HTML OBJECT or APPLET tags.
"E"	The link is from a cascading style sheet (CSS) .
"F"	The link is from the SRC attribute of an HTML FRAME tag.
"G"	The link is to a dynamic Web template for the containing document.
"H"	The link is from an HTML HREF attribute. This can also be used as a default link type value if a more precise type does not apply.
"I"	The link is to a document that the containing document includes via an include bot.
"J"	The link is from a field (1) of this list item.
"K"	Identical to "H", except that the link also specifies an HTML bookmark.
"L"	The link is a target in an HTML image map generated by a bot.
"M"	The link is to an image used in an HTML image map generated by a bot.
"O"	The link is part of a cross-page URL connection.
"P"	The link is part of the markup of a URL within the source of the containing document.
"Q"	The link references a cascading style sheet (CSS) document that provides style information for the containing document.
"R"	The link is from the master page file attribute of the @Page directive in the containing document.
"S"	The link is from an HTML SRC attribute.
"T"	The link is to the index file used by a text search bot on this page.
"V"	The link is based on the properties of the document, rather than anything in the document stream. This link type is used in tracking the link between a site and the master page URL used for the site.
"X"	The link is from an XML island within an HTML document.
"Y"	The link references an HTML document whose HTML BODY tag attributes are used as a template for the attributes of the containing document's BODY tag.
"Z"	The link is part of the markup of a URL that exists in a URL zone in the containing document, and is consequently not stored within the source of the containing document.

@Security: Type of security for the link. The value **MUST** be one of the values in the following table.

Value	Description
"H"	The link is to an HTTP URL.
"S"	The link is to an HTTPS URL.
"T"	The link is to an SHTTP URL.
"U"	The link transport security is unknown.

@Dynamic: A 1-byte (**tinyint**) value represented as a single upper case ASCII character which specifies the special link types. The value MUST be one of the values in the following table.

Value	Description
"S"	The URL is static, which is the default, and requires no special handling.
"D"	The URL is dynamic, which is a link to <Site URL>/_vti_bin/shtml.dll/DirName/LeafName. Such links are used to call the SmartHTML interpreter on a file.
"L"	The URL is to a layouts page; that is, it contains a path segment with the string "_layouts".
"H"	The URL is a history link ; that is, it contains a path segment with the string "_vti_history".
"G"	A link that is not absolute from an uncustomized document that does not fall into any other category.

@ServerRel: A bit flag that specifies whether the link URL is a **server-relative URL**. A value of "1" specifies a server-relative URL. This value MUST NOT be NULL.

@Search: Search terms to be used to surface this link when performing a full-text search. This value can be an empty string, but it MUST NOT be NULL.

@WP: A **Web Part identifier** identifying the Web Part that is the source of the link creation command.

@Fld: The field identifier of the field that is the source of the link definition.

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

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

Result sets: MUST NOT return any result sets.

3.1.4.9 proc_AppendFileFragmentDataById

The **proc_AppendFileFragmentById** stored procedure is called to append data for an existing file fragment. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_AppendFileFragmentDataById (
    @SiteId          uniqueidentifier,
    @DocId           uniqueidentifier,
    @DocVersion      int,
    @UserId          int,
    @Partition       tinyint,
    @Id              bigint,
    @BlobData        varbinary(max),

```

```

        @BlobSize          int
    );

```

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

@DocId: The document identifier (2) of the document associated with the file fragment being updated.

@DocVersion: This parameter MUST be ignored.

@UserId: This parameter MUST be ignored.

@Partition: The Identifier for a file fragment partition of the **partition (2)** to which the file fragment being updated belongs.

@Id: The **file fragment identifier** of the file fragment to update.

@BlobData: The data to append to the existing data of the file fragment.

@BlobSize: The size in bytes of the data being appended.

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

Value	Description
0	Successful execution.
33	The file fragment being updated with values @DocId, @Partition, @Id could not be found.
4317	The file fragment data being updated resulted in an IO error or constraint violation.

Result Sets: MUST NOT return any result sets.

3.1.4.10 **proc_CascadeDeleteItems**

The **proc_CascadeDeleteItems** stored procedure is called to perform a cascading delete operation. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_CascadeDeleteItems (
    @SiteId          uniqueidentifier,
    @WebId           uniqueidentifier,
    @ParentListId   uniqueidentifier,
    @ParentItemId   int,
    @ParentUrl      nvarchar(260),
    @ParentServerTemplate int,
    @ParentBaseType int,
    @ParentEventData varbinary(max),
    @ParentAcl      varbinary(max),
    @ParentNeedsAuthorRestriction bit,
    @ParentListDeletedUrls bit,
    @ParentListItemVersion bit,
    @UserTitle      nvarchar(255),
    @UserId         int,
    @QueryAuditFlags bit,
    @LogChange      bit,
    @ThresholdRowCount int,

```

```

@CsvListItemString          nvarchar(max),
@DeleteOp                   int,
@FailedUrl                  nvarchar(260) = null OUTPUT,
@DeleteTransactionId        varbinary(16) = 0x OUTPUT
);

```

@SiteId: The site collection identifier of the site collection containing the specified list (1).

@WebId: The site identifier of the site (2) containing the specified list (1).

@ParentListId: The list identifier of the list (1) containing the parent item of the cascading delete operation.

@ParentItemId: The list item identifier of the parent item of the cascading delete operation.

@ParentUrl: The store-relative form Uniform Resource Locator (URL) of the parent item of the cascading delete operation.

@ParentServerTemplate: The identifier for the **list template** that defines the base structure of the specified list (1).

@ParentBaseType: The **base type** of the list (1) containing the parent item of the cascading delete operation. It MUST NOT be NULL and MUST be one of the values specified in [\[MS-WSSTS\]](#) section 2.5.

@ParentEventData: The parameter contains implementation-specific event data significant to the front-end Web server but otherwise opaque to the back-end database server, to be stored by the back-end database server for eventual writing to the change log.

@ParentAcl: The binary serialization of the Windows SharePoint Services ACL Format ACL for the data supplied in **@ParentEventData**. It can be NULL. For more information regarding this ACL Format, see [\[MS-WSSF02\]](#) section 2.2.4.6.

@ParentNeedsAuthorRestriction: A bit flag to specify whether to verify if the **current user** specified by **@UserId** is the **author** of the parent item of the cascading delete operation. If this is 1 and the current user specified by **@UserId** is not the author of the parent item of the cascading delete operation, the stored procedure MUST NOT complete successfully. If **@ParentBaseType** is 1, this bit MUST be set to NULL.

@ParentListDeletedUrls: A bit flag to specify whether the **Deleted Documents Result Set** (section [3.1.4.10.1](#)) MUST be returned as a result of successfully deleting the parent item of the cascading delete operation. If the **Deleted Documents Result Set** MUST be returned, this MUST be set to 1, otherwise this MUST be set to zero. If **@ParentBaseType** is not 1, this bit MUST be set to zero.

@ParentListItemVersion: An optional value to compare with the **version** number of the list item. It can be NULL. If this parameter is not NULL, the parameter MUST match the version number for successful completion.

@UserTitle: The display name of the current user. This parameter MUST be ignored.

@UserId: The identifier of the current user. This value MUST NOT be NULL. This value MUST be the user identifier of a user that belongs to the specified site collection.

@QueryAuditFlags: A bit flag to specify whether an **audit entry** MUST be created for the cascading delete operation. If this is 1, an audit entry MUST be created for the cascading delete operation. If this is zero, an audit entry MUST NOT be created for the cascading delete operation.

@LogChange: A bit flag to specify whether the change log MUST be updated. If this is 1, the change log MUST be updated. If this is zero, the change log MUST NOT be updated.

@ThresholdRowCount: If the current user has permissions to ignore the throttle restrictions, this MUST be zero; otherwise this MUST be the throttling limit. If the value is not zero, then the stored procedure MUST NOT complete the cascading delete operation successfully and return the error code 36 if the number of documents deleted by the deletion of any child item or the parent item in the cascading delete operation exceeds the value of this parameter.

@CsvListItemString: A parameter specifying the collection of child items to be deleted in the cascading delete operation. This parameter MUST conform to the following ABNF:

CsvListItemString = ListAndListItems 0*("," \$ "," ListAndListItems)

ListAndListItems = ListId 1*("," ListItemId)

ListId = GUID

ListItemId = 1*DIGIT

GUID = "{" GUIDCORE "}" / GUIDCORE

GUIDCORE = 8HEXDIG "-" 4HEXDIG "-" 4HEXDIG "-" 4HEXDIG "-" 12HEXDIG

The **ListId** MUST specify the list identifier containing the child item of the cascading delete operation. The **ListItemId** MUST specify the list item identifier of the child item and MUST be greater than zero and less than 2,147,483,648.

@DeleteOp: A value specifying the type of delete operation to attempt. The value MUST be one of the values listed in the following table.

Value	Description
3	Deleting the child items and parent item of the cascading delete operation without placing them in the Recycle Bin.
4	Deleting the child items and parent item of the cascading delete operation and placing them in the Recycle Bin. This value MUST be set only if the Recycle Bin is available.

@FailedUrl: An output parameter indicating the URL at which the delete operation failed. This parameter MUST be set to NULL if the deletion was successful.

@DeleteTransactionId: An output parameter used to identify all list items, documents or folders deleted as part of the cascading delete operation. If **@DeleteOp** is 4, **@DeleteTransactionId** MUST be set to a unique delete transaction identifier. Otherwise, it MUST be ignored.

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

Value	Description
0	Successful execution.
3	The parent item or one of the child items of the cascading delete operation could not be found.

Value	Description
5	The current user specified by @UserId is not same as the author of the parent item of the cascading delete operation when @ParentNeedsAuthorRestriction is set to 1.
33	While deleting the parent item of the cascading delete operation, an attempt is made to delete folders that contain checked out files.
36	The number of rows deleted by deleting any child item or the parent item of the cascading delete operation exceeds @ThresholdRowCount .
1150	Concurrency violation. The @ParentListItemVersion parameter does not match the version number of the parent item of the cascading delete operation. This value MUST only be returned if the @ParentListItemVersion parameter is not NULL.
1359	An internal error occurred, or bad parameter was specified.
6009	While deleting the parent item of the cascading delete operation, an attempt is made to delete folders that contain files that are in read-only mode .

Result sets: MUST return the **Deleted Documents Result Set** (section [3.1.4.10.1](#)) when **@ParentListDeletedUrls** is set to 1 and if the deletion of the parent item of the cascading delete operation finished successfully.

3.1.4.10.1 Deleted Documents Result Set

The **Deleted Document Result Set** MUST be returned only for the parent item of the cascading delete operation when **@ParentListDeletedUrls** is set to 1 and the deletion of the parent item finished successfully. For more information, see [\[MS-WSSFO2\]](#) section 3.1.5.13.1.

3.1.4.11 proc_CheckIfExistingFieldHasDuplicateValues

The **proc_CheckIfExistingFieldHasDuplicateValues** stored procedure is called to verify whether there are at least two list items in a list (1) which, in their most recent published or draft versions, have the same value for a specified field (1). The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_CheckIfExistingFieldHasDuplicateValues (
    @SiteId                uniqueidentifier,
    @WebId                 uniqueidentifier,
    @ListId                uniqueidentifier,
    @FieldId               uniqueidentifier,
    @Collation              smallint
);

```

@SiteId: The site collection identifier of the site collection containing the specified list (1).

@WebId: The site identifier of the site (2) containing the specified list (1).

@ListId: The list identifier of the specified list (1).

@FieldId: The field identifier of the specified field (1).

@Collation: The collation order for the site (2) that contains this list (1). If the Field Internal Type Name, as defined in [\[MS-WSSTS\]](#), Field Type and Field Internal Type, as specified in [\[MS-WSSTS\]](#)

section 2.1.2.9.1, is "Text" or "Choice", it MUST NOT be NULL and MUST be one of the collation order enumeration values specified in [\[MS-WSSFO2\]](#) section 2.2.3.4. Otherwise, it MUST be NULL.

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

Value	Description
0	If the collation order is NULL, there are no two list items in the list (1) which, in their most recent published or draft versions, have the same non-empty value for the specified field (1). Otherwise, if the collation order is not NULL, there are no two list items in the list (1) which, in their most recent published or draft versions, have the same non-empty value for the specified field (1), where equality is defined by the specified collation order.
2	If the collation order is NULL, there are at least two list items in the list (1) which, in their most recent published or draft versions, have the same non-empty value for the specified field (1). Otherwise, if the collation order is not NULL, there are at least two list items in the list (1) which, in their most recent published or draft versions, have the same non-empty value for the specified field (1), where equality is defined by the specified collation order.

Result sets: MUST NOT return any result sets.

3.1.4.12 **proc_CheckIfExistingLookupsHaveValidParents**

The **proc_CheckIfExistingLookupsHaveValidParents** stored procedure is called to verify whether there is at least one list item in a list (1) which, in its most recent published or draft version, contains a value for a **lookup field** which is not the list item identifier of an existing list item in the specified target list (1) of the specified lookup field. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_CheckIfExistingLookupsHaveValidParents (
    @SiteId          uniqueidentifier,
    @ListId           uniqueidentifier,
    @FieldId         uniqueidentifier,
    @LookupListId    uniqueidentifier,
    @RequestGuid     uniqueidentifier = null OUTPUT
);

```

@SiteId: The site collection identifier of the site collection containing the specified list (1).

@ListId: The list identifier of the specified list (1).

@FieldId: The field identifier of the specified lookup field.

@LookupListId: The target list (1) of the specified lookup field.

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

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

Value	Description
0	There is no such list item in the list (1) which, in its most recent published or draft version, contains a value for the specified lookup field which is not the list item identifier of an existing list item in the list (1) specified by @LookupListId.
3	There is at least one list item in the list (1) which, in its most recent published or draft version,

Value	Description
	contains a value for the specified lookup field which is not the list item identifier of an existing list item in the list (1) specified by @LookupListId.

Result sets: MUST NOT return any result sets.

3.1.4.13 proc_CheckIfRestoreNeedsTranLock

The **proc_CheckIfRestoreNeedsTranLock** stored procedure is called to verify whether a short-term transaction application lock MUST be set on the site (2) of a **Recycle Bin item**. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_CheckIfRestoreNeedsTranLock (
    @SiteId                uniqueidentifier,
    @WebId                 uniqueidentifier,
    @UserId                int,
    @DeleteTransactionId   varbinary(16),
    @TranLockStatus        int OUTPUT,
    @TranLockWebIdToUse   uniqueidentifier OUTPUT,
    @RequestGuid           uniqueidentifier = null OUTPUT
);

```

@SiteId: The site collection identifier of the site collection which contains the Recycle Bin item with the specified delete transaction identifier.

@WebId: The site identifier of the site (2) which contains the Recycle Bin item with the specified delete transaction identifier.

@UserId: The user identifier of the current user, or zero for administrative access to the Recycle Bin of any user.

@DeleteTransactionId: The delete transaction identifier of the Recycle Bin item.

@TranLockStatus: The value that specifies whether a short-term **transaction application lock** was taken for the specified site (2). This is an output parameter and MUST specify a valid value from the following table.

Value	Description
0	The stored procedure did not complete successfully.
1	The short-term transaction application lock is not needed. This MUST be returned when either of the following conditions are true: <ul style="list-style-type: none"> ▪ The Recycle Bin item is a document version. ▪ The Recycle Bin item is a list (1). ▪ The Recycle Bin item is a folder containing lists (1). ▪ The Recycle Bin item is an attachment. ▪ The Recycle Bin item is a document, list item or folder and the following are true: <ul style="list-style-type: none"> ▪ The list (1) containing the Recycle Bin item has no relationship lookup fields with

Value	Description
	<p>cascading behavior or restrict behavior.</p> <ul style="list-style-type: none"> There is no relationship lookup field in any list (1) in the site (2) which has the list (1) containing the Recycle Bin item as the target list (1) and has cascading behavior or restrict behavior.
2	<p>The short-term transaction application lock is needed. This MUST be returned when either of the following conditions are true:</p> <ul style="list-style-type: none"> The Recycle Bin item is the parent item of a cascading delete operation. The Recycle Bin item is a document, list item or folder and the following are true: <ul style="list-style-type: none"> The list (1) containing the Recycle Bin item has at least one relationship lookup field with cascading behavior or restrict behavior. There is at least one relationship lookup field in some list (1) in the site (2) which has the list (1) containing the Recycle Bin item as the target list (1) and has cascading behavior or restrict behavior.

@TranLockWebIdToUse: The site identifier of the site (2) for which the short-term transaction application lock was taken. This is an output parameter.

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

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

Value	Description
0	Successful Execution.
1168	No Recycle Bin item is found for @SiteId AND @DeleteTransactionID when @UserId is zero; OR no Recycle Bin item is found for @SiteId, @WebId, @DeleteTransactionID AND @UserId when @UserId is NOT zero; OR more than one Recycle Bin item is found for the given parameters.
1359	An internal error has occurred.

Result sets: MUST NOT return any result sets.

3.1.4.14 **proc_CheckoutDocumentInternal**

The **proc_CheckoutDocumentInternal** stored procedure is called to request or renew short-term check-out, or to request long-term check-out on a document. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_CheckoutDocumentInternal (
    @SiteId                uniqueidentifier,
    @WebId                 uniqueidentifier,
    @DirName               nvarchar(256),
    @LeafName              nvarchar(128),
    @Level                 tinyint,
    @EnableMinorVersions   bit,
    @IsModerated           bit,
    @UserId                int,
    @CheckoutTimeout       int,

```

```

@RefreshLock          bit,
@CheckoutToLocal     bit,
@IsForceCheckout     bit,
@IsSharedLock        bit,
@IsConvertLock       bit,
@Now                 datetime
@DocMetaInfo         varbinary(max),
@DocMetaInfoSize     int,
@DocMetaInfoVersion  int,
@RequestGuid         uniqueidentifier = NULL OUTPUT
);

```

@SiteId: The site collection identifier of the site collection which contains the document to be checked out.

@WebId: The site identifier of the site (2) which contains the document.

@DirName: The directory name of the document.

@LeafName: The leaf name of the document.

@Level: The publishing level of the document. For valid values, see [\[MS-WSSFO2\]](#) section 2.2.2.6.

@EnableMinorVersions: A bit flag specifying whether the document library containing the document has **minor version** numbering enabled. If minor version numbering is enabled for the document library containing the document, this parameter **MUST** be set to 1; otherwise this parameter **MUST** be set to zero. If the document is not in a document library, this parameter **MUST** be set to zero. This parameter **MUST NOT** be NULL.

@IsModerated: A bit flag specifying whether the document library containing the document has **moderation** enabled. If the document library containing the document is a **moderated object**, this parameter **MUST** be set to "1"; otherwise this parameter **MUST** be set to "0". If the document is not in a document library, this parameter **MUST** be set to "0". This parameter **MUST NOT** be NULL.

@UserId: The user identifier for the current user who is requesting a short-term check-out or a long-term check-out on the document. This value **MUST** refer to an existing user identifier for the specified site collection.

@CheckoutTimeout: The time-out in minutes for short-term check-out on the document. The **@CheckoutTimeout** parameter **MUST** be NULL if a long-term check-out on the document is being specified.

@RefreshLock: A bit flag specifying whether the short-term check-out on the document needs to be refreshed. If this parameter is set to 1, the existing short-term check-out on the document **MUST** be refreshed for the number of minutes specified by the **@CheckoutTimeout** parameter. This parameter **MUST** be set to zero to request a new short-term check-out or long-term check-out on the document. This parameter **MUST NOT** be NULL.

@CheckoutToLocal: A bit flag specifying whether the document is to be copied to local storage on the user's computer for editing. If this parameter is set to 1, the User computer **SHOULD** make a local copy of the document stream for editing and **proc_CheckoutDocumentInternal** **MUST NOT** make a checked-out version of the document in the back-end database servers .

@IsForceCheckout: A bit flag specifying whether the document library containing the document requires that documents be checked out before any changes to the document can be made. If check-out is required for the document library containing the document, this parameter MUST be set to 1; otherwise this parameter MUST be set to zero. If the document is not in a document library, this parameter MUST be set to zero. This parameter MUST NOT be NULL.

@IsSharedLock: A bit flag specifying whether the desired short-term lock on the document is a shared lock or an exclusive lock. This parameter MUST be set to 1 if the desired short-term lock is type shared; otherwise it MUST be set to zero.

@IsConvertLock: A bit flag specifying whether to convert an existing short-term lock from one short-term lock type to a different short-term lock type. This parameter MUST be set to 1 to convert the type of an existing short-term lock; otherwise it MUST be set to zero.

@Now: The current Coordinated Universal Time (UTC) time.

@DocMetaInfo: The metadata information for the document to be checked out. If there is no metadata information for this document, this parameter MUST be NULL.

@DocMetaInfoSize: Size in bytes of the document's metadata info. This MUST be NULL if @DocMetaInfo is NULL.

@DocMetaInfoVersion: The version of the metadata information for the document to be checked out. This MUST be NULL if @DocMetaInfo is NULL.

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

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

Value	Description
0	Successful execution.
3	File not found. A document corresponding to the specified @SiteId, @WebId, @DirName, @LeafName, and @Level parameters was not found.
33	Short-term lock error. The document cannot have a short-term lock applied because another user has the document checked out.
154	Invalid minor version value. The minor version value for the document would exceed the maximum allowed value (511) if the document was checked out.
158	Check-out required. The document is in a document library with the Require Check Out option set, but the document is not checked out.
212	Site collection locked. The operation could not be performed because the site collection containing the document is in read-only mode.
1460	The lock has expired or the requested operation cannot be performed on the lock.
1630	Unsupported document type. The document specified is not valid for check-out; folders and sites (2) cannot be checked out.
1816	Disk quota error. The site collection disk quota has been reached.
6002	Short-term lock error. The document cannot have a short-term lock applied because another user has a shared short-term lock on the file.
6009	Short-term lock error. The document cannot have a short-term lock applied because

Value	Description
	another user has an exclusive short-term lock on the file.

Result sets: MUST NOT return any result sets.

3.1.4.15 `proc_CloneDoc`

The **proc_CloneDoc** stored procedure is called to create a copy or a new version of an existing document. The target document has exactly the same contents and properties as that of the original document except publishing level and version. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_CloneDoc (
    @SiteId          uniqueidentifier,
    @DirName         nvarchar(256),
    @LeafName        nvarchar(128),
    @NewInstanceID  int = NULL,
    @NewItemID      int = NULL OUTPUT,
    @Now             datetime = NULL,
    @OldLevel        int = NULL,
    @NewLevel        int = NULL,
    @EnableMinorVersions bit = NULL,
    @IsModerated    bit = NULL,
    @UserId          int = NULL,
    @NewLeafName     nvarchar(128) = NULL,
    @RequestGuid     uniqueidentifier = NULL OUTPUT
);

```

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

@DirName: The directory name containing the existing document.

@LeafName: The leaf name of the existing document.

@NewInstanceID: If the specified document is inside a **Meeting Workspace site**, this parameter MUST be the identifier of the meeting workspace site. Otherwise, the value can be omitted and it defaults to NULL.

@NewItemID: If the target document is successfully created, **proc_CloneDoc** MUST return the identifier of the target document. If the operation fails, this value MUST be ignored.

@Now: The current Coordinated Universal Time (UTC) time. If the value is omitted it defaults to NULL.

@OldLevel: A publishing level specifying the publish status of the existing document. If the value is omitted, it defaults to NULL. If the value equals **@NewLevel** or NULL, a new document MUST be created.

@NewLevel: The publishing level of the target document or version. If the value is omitted, it defaults to NULL. If the value equals **@OldLevel** or NULL, a new document MUST be created.

@EnableMinorVersions: A bit flag specifying whether the document library containing the existing document has minor version numbering enabled. If minor version numbering is enabled for the document library containing the existing document, this parameter MUST be set to 1; otherwise,

this parameter MUST be set to zero. If the document is not in a document library, this parameter MUST be set to zero. This parameter MUST NOT be NULL.

@IsModerated: A bit flag specifying whether the document library containing the existing document has **moderation** enabled. If the document library containing the existing document is a moderated object, this parameter MUST be set to 1; otherwise, this parameter MUST be set to zero. If the document is not in a document library, this parameter MUST be set to zero. This parameter MUST NOT be NULL.

@UserId: The identifier for the current user who is requesting this operation. The value MUST be provided if **@NewInstanceID** is NULL, or if the values of **@OldLevel** is not the same as **@NewLevel**. If a value is provided, the value MUST refer to an existing user identifier for the specified site collection. If the value is omitted, it defaults to NULL.

@NewLeafName: The leaf name of the target document or version. If the value is omitted, it defaults to NULL.

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

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

Value	Description
0	Successful execution. The document was successfully copied.
3	File Not Found. The target document corresponding to the specified @SiteId , @WebId , @DirName , and @NewLeafName could not be created.
87	Invalid Parameter. One or more of the parameters, @SiteId , @WebId , @DirName , @LeafName , or @NewLevel , are not valid.
160	Incorrect parameters. The value of @NewLevel is "Draft" and @UserID is NULL.
212	Site collection locked. The operation could not be performed because the site collection containing the existing document is in read-only mode.
1816	Disk quota error. The quota for the site collection has reached the maximum allowable limit.

Result sets: MUST NOT return any result sets.

3.1.4.16 **proc_ConvertJunctionToLookup**

The **proc_ConvertJunctionToLookup** stored procedure is called to convert the type of a lookup field of a list (1) from **multivalued lookup field** to single-value lookup field. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ConvertJunctionToLookup(  
    @SiteId                uniqueidentifier,  
    @ListId                uniqueidentifier,  
    @FieldId              uniqueidentifier,  
    @ColName               nvarchar(64),  
    @RowOrdinal            int,  
    @RequestGuid           uniqueidentifier = NULL OUTPUT  
);
```

@SiteId: The site collection identifier of the site collection which contains the specified list (1).

@ListId: The list identifier of the list (1) containing the specified lookup field.

@FieldId: The field identifier of the specified lookup field.

@ColName: The name of the **column (1)** in the [AllUserData Table](#) which corresponds to the specified lookup field.

@RowOrdinal: The row ordinal of the lookup field.

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

Return values: An integer that MUST be zero.

Result sets: MUST NOT return any result sets.

3.1.4.17 `proc_ConvertLookupToJunction`

The **proc_ConvertLookupToJunction** stored procedure is called to convert the type of a lookup field of a list (1) from single-value lookup field to multivalued lookup field. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ConvertLookupToJunction(  
    @SiteId            uniqueidentifier,  
    @ListId            uniqueidentifier,  
    @FieldId           uniqueidentifier,  
    @ColName           nvarchar(64),  
    @RowOrdinal        int,  
    @RequestGuid       uniqueidentifier = NULL OUTPUT  
);
```

@SiteId: The site collection identifier of the site collection which contains the specified list (1).

@ListId: The list identifier of the list (1) containing the specified lookup field.

@FieldId: The field identifier of the specified lookup field.

@ColName: The name of the column in the [AllUserData Table](#) which corresponds to the specified lookup field.

@RowOrdinal: The row ordinal of the specified lookup field.

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

Return Code Values: An integer that MUST be zero.

Result Sets: MUST NOT return any result sets.

3.1.4.18 `proc_CopyUrl`

The **proc_CopyUrl** stored procedure is called to copy a site collection, or **subsite** to a new location specified by a new URL. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_CopyUrl(  
    @SiteId            uniqueidentifier,  
    @SubWebId          uniqueidentifier,  
    @OldUrl            nvarchar(260),  
    @NewUrl            nvarchar(260),  
    -
```

```

@UserId                int,
@TresholdRowCount      int,
@NewDoclibRowIdInput  int,
@MaxNewRowsInput      int,
@RenameFlags          int = 0,
@PutFlags              int = 0,
@ReturnFlags          int = 0,
@AttachmentOp         int = 3,
@ParseDocsNow         tinyint          = NULL OUTPUT,
@FailedUrl            nvarchar(260) = NULL OUTPUT,
@RequestGuid          uniqueidentifier = NULL OUTPUT
);

```

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

@SubWebId: A site identifier that uniquely identifies the subsite. MUST NOT be NULL.

@OldUrl: Current URL of the item being copied.

@NewUrl: URL which is to replace the **@OldUrl**.

@UserId: User identifier of the requester.

@ThresholdRowCount: If the requester has permissions to ignore the throttling restrictions, this MUST be zero, else this MUST specify the max number of documents the container represented by **@OldUrl** can have. If there are more than the **@ThresholdRowCount** documents in the container, the stored procedure MUST return the error code of 36 and fail the operation. It MUST be zero or the same as the throttling limit.

@NewDoclibRowIdInput: The next available item identifier in the new list (1) where this item is being copied. This value is obtained by the call to **proc_GetNewListItemId** (section [3.1.4.61](#))

@MaxNewRowsInput: The number of items that are being copied.

@RenameFlags: A 4-byte integer bit mask determining the object rename options. This can have one or more flags set. The default value is zero, but it MUST NOT be NULL. The valid flags are described in the following table.

Value	Description
0x00000000	Default behavior: Rename all dependent items.
0x00000001	Do not update all related documents.
0x00000002	Create directories if they do not exist.
0x00000004	Server MUST find backward links to rename them and update the original document.
0x00000008	Return thicket folders or files.
0x00000010	Fix links within the same URL subtree. Used when doing link fixup after a directory has been renamed.
0x00000020	Allow renaming of sites (2).
0x00000040	Allow the setting of the "CanBeParsed" document flag when a file's extension changes.
0x00000080	Allow update of the "CanHaveLinks" document flag when a file's extension changes.

Value	Description
0x00000100	Allow renaming of sites (2) and directory names.
0x00000200	Allow move into the forms directory.
0x00000400	Current user can view draft documents.
0x00000800	Allow move operation on a thicket with missing thicket supporting files .

@PutFlags: A 4-byte integer bit mask determining document change options. This can have one or more flags set. The default value is zero, but it MUST NOT be NULL. The valid flags are described in the following table.

Value	Meaning
0x00000008	Keep the document checked out.
0x00000020	Check in the document.
0x00001000	Create a new displayed version of the document, even if it is in a short-term check-out.
0x00002000	Use client metadata for user, date and time for creation, last modification, and check-in comments.
0x00010000	Publish the document.
0x00020000	Overwrite the document without updating its displayed version.
0x00100000	The document is being added or updated as part of a system update. Do not update the last modification time and user.
0x00800000	Do not increment the internal version number for the document. This flag SHOULD be set only if the user can tolerate having their changes overwritten by another user in the event of a conflict.
0x02000000	Keep the document checked out to the user's local disk.

@ReturnFlags: A 4-byte integer bit mask determining the result sets returned from stored procedures called by **proc_CopyUrl**. This can have one or more flags set. The default value is zero, but it MUST NOT be NULL. The valid flags are in the following table.

Value	Description
0x00	Return no result set data.
0x01	Return result sets pertaining to renamed documents
0x02	Return result sets pertaining to moved documents with patched links.

@AttachmentOp: An integer value which governs the type of security checks that SHOULD be performed by the stored procedure on this document's URL based on whether it appears to be an attachment. The integer value MUST be listed in the following table.

Value	Description
0	Document is not an attachment. Do not perform attachment flag update.

Value	Description
1	Update the item's flags only.
2	Update the item's version. In addition to performing an update on the attachment flag, the version number for the attachment is to be updated along with the timestamp for when the document was modified, the publishing level, and the editor of the document. This flag is set only if the user requesting the update has permissions to modify the list (1).
3	Update the item's modification state. In addition to performing an update on the attachment flag, the modification timestamp, the level , and the associated editor of the attachment is to be updated.

@ParseDocsNow: Bit indicator that a document needs further information gathering. Whenever a document is moved into a new document library, the metadata needs to be updated and **@ParseDocsNow** is set to 1. If the document is left in the same document library or moved within a document library, **@ParseDocsNow** MUST be set to zero or NULL.

@FailedUrl: If a delete, copy, or move operation fails because of invalid parameters or permissions, this is filled in with the **site-relative URL** for the specific failed document.

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

Value	Description
0	Successful execution.
2	The specified destination was not found.
3	The specified site (2) or subsite was not found.
5	User is not authorized to make this change.
15	Attempt to rename an excluded directory type.
32	There was a sharing or lock violation.
33	Attempted to move directories that contain checked out files.
36	There are more than the @ThresholdRowCount documents in the container represented by @OldUrl .
50	Attempt to rename a site (2) inside a list (1).
51	Attempted to rename a Forms folder .
53	There is an inconsistency between the specified and the expected value of @SubWebId . The only way this happens is if there are concurrent attempts made to change affected objects.
80	Invalid @PutFlags for a file operation.
87	There is an inconsistency between the expected number of documents to be modified and the observed number which would be modified. The only way this happens is if there are concurrent attempts made to change affected objects.
130	Attempted to rename the thumbnail or image part of a thicket.
138	Attempted to copy folders that span lists (1).

Value	Description
144	Old and new URL object types are not the same.
161	Attempted to copy folders that span sites (2).
190	Attempted to create a thicket.
206	Attempted to move folders that exceed file name range.
212	Write Lock Error when creating a file or directory.
214	Attempted to copy a thicket.
266	Specified old and new URL are the same.
1150	Concurrency violation.
1359	Internal error occurred.
1816	Disk quota exceeded
8389	At least one of the lists (1) could not be deleted.

Result sets: The stored procedure returns zero or more result sets depending on conditions described in the result set. Some of the result sets are returned zero or more times depending upon input parameters and type of URL to be copied. All result sets that are returned MUST be returned in the order listed.

3.1.4.18.1 NULL List Metadata Result Set

If the copied URL has no lists (1), this MUST be returned zero, one or two times. It is returned twice if the **@ReturnFlags** does not have the 0x01 bit set; once for the old URL, and once for the new URL. If the **@ReturnFlags** does have the 0x01 bit set, a **NULL List Metadata Result Set** will be returned if the old URL location does not have a containing list (1), or if the new URL location does not have a containing File result set. This result set MUST be returned if the specified object is a file. It returns a log of the old file name and the new file names including the directory path information. The T-SQL syntax for the result set is as follows:

```
{OldUrlDirName}          nvarchar (256),
{OldUrlLeafName}         nvarchar (128),
{NewUrlDirName}          nvarchar (256),
{NewUrlLeafName}         nvarchar (128),
{Type}                   int;
```

{OldUrlDirName}: Directory path as it exists before any transformation takes place.

{OldUrlLeafName}: The file name before any transformation takes place.

{NewUrlDirName}: Directory path as it exists after any transformation takes place.

{NewUrlLeafName}: The file name after any transformation takes place.

{Type}: Type of URL being transformed. This value MUST be zero.

3.1.4.18.2 Copied Directory Result Set

The **Copied Directory Result Set** MUST be returned if the specified object is a directory. It returns a log of the old directory name and new directory name including directory path information. The T-SQL syntax for the result set is as follows:

OldDirName	nvarchar(256),
OldLeafName	nvarchar(128),
NewDirName	nvarchar(256),
NewLeafName	nvarchar(128),
Type	int;

OldDirName: Directory path as it exists before any transformation takes place.

OldLeafName: The directory name before any transformation takes place.

NewDirName: Directory path as it exists after any transformation takes place.

NewLeafName: The directory name after any transformation takes place.

Type: Type of URL being transformed. This value MUST be 1.

3.1.4.19 proc_CreateList

The **proc_CreateList** stored procedure is called to create a new entry in the content database for the specified list (1) and to return its metadata and **full URL**. The stored procedure is defined using T-SQL syntax, as follows:

```
PROCEDURE proc_CreateList(
    @SiteId                uniqueidentifier,
    @WebId                 uniqueidentifier,
    @ListId                uniqueidentifier,
    @DirName               nvarchar(256),
    @FolderNameBase        nvarchar(50),
    @bAlternateUrlOnCollision bit,
    @Title                 nvarchar(255),
    @Version               int,
    @Author                int,
    @BaseType              int,
    @bCreateAttachmentsSubFolder bit,
    @FeatureId             uniqueidentifier,
    @ServerTemplate         int,
    @DocLibTemplate         uniqueidentifier,
    @ImageUrl              nvarchar(255),
    @SendToLocation        nvarchar(512),
    @ReadSecurity           int,
    @WriteSecurity         int,
    @Description            nvarchar(max),
    @MajorVersionCount     int,
    @MinorVersionCount     int,
    @Fields                 varbinary(max),
    @Direction             int,
    @Flags                 bigint,
    @ThumbnailSize         int,
    @WebImageWidth         int,
    @WebImageHeight        int,
    @bParentFolderChecked bit,
```

```

@OnRestore                bit,
@EventSinkAssembly        nvarchar(255),-v_store
@EventSinkClass           nvarchar(255),
@EventSinkData            nvarchar(255),
@ContentTypes             varbinary(max),
@RootFolderId            uniqueidentifier = NULL,
@TitleResource            nvarchar(256) = NULL,
@DescriptionResource      nvarchar(256) = NULL,
@ValidationFormula        nvarchar(1024) = NULL,
@ValidationMessage        nvarchar(1024) = NULL,
@FolderFullUrlRet        nvarchar(256) = NULL OUTPUT,
@TimeCreated              datetime = NULL,
@RequestGuid              uniqueidentifier = NULL OUTPUT
);

```

@SiteId: The site collection identifier of the site collection which contains the specified site (2).

@WebId: The site identifier of the site (2) which contains the specified list (1).

@ListId: The list identifier of the list (1) that is being created.

@DirName: The directory name of the list (1) location.

@FolderNameBase: A unique directory name used for the list (1).

@bAlternateUrlOnCollision: A bit that specifies whether or not to generate a unique alternate directory name specified by the **@FolderNameBase** parameter in the case of a name collision.

@Title: The title of the specified list (1).

@Version: The initial list (1) version to begin with.

@Author: The user identifier of the list (1) author.

@BaseType: This specifies the base type of the list (1). The value MUST be one of the following list (1) base types.

Value	Description
0	Generic List
1	Document Library
3	Discussion Board
4	Survey List
5	Issues List

@bCreateAttachmentsSubFolder: A bit specifying whether or not to create a subfolder for list (1) attachments.

@FeatureId: The **feature identifier** of the feature associated with the list (1).

@ServerTemplate: The integer value of the list template that defines the base structure of this list (1).

@DocLibTemplate: The list template of the document library.

@ImageUrl: Contains the server-relative URL that points to an image associated with the list (1).

@SendToLocation: Contains an implementation-specific Send-To string holding a URL used for the list (1). If this parameter is NULL, it MUST be ignored.

@ReadSecurity: A value identifying the security policy for read access on list items. If value is set to 1, users with read permissions can read all list items. Otherwise, if this value is set to 2, users with read permissions can only read their own list items.

@WriteSecurity: A value identifying the security **policy** for write access on list items. The value MUST be in the following table.

Value	Description
1	Users with write permissions have write access to all list items.
2	Users with write permissions have write access to their own list items only.
4	Users have no write access to any list items.

@Description: The text describing the list (1).

@MajorVersionCount: Sets the maximum number of **major versions** to be retained by the list (1).

@MinorVersionCount: Sets the maximum number of minor versions to be retained by the list (1).

@Fields: A version string followed by the XSD representation of the **field definitions**. The field definitions include display and interaction options. See [\[MS-WSSFO2\]](#) section 2.2.8.3.5.

@Direction: An enumerated value specifying the direction of text flow for user interface elements presented by this list (1). The value MUST be in the following table.

Value	Description
0	No explicit direction is specified.
1	Text flow SHOULD be left to right.
2	Text flow SHOULD be right to left.

@Flags: A bit array for setting list (1) functionality. This parameter MUST not be NULL. Valid values are contained in the table defined in [\[MS-WSSFO2\]](#) section 2.2.2.5.

@ThumbnailSize: An integer used by lists (1) to determine the rendering size of an image thumbnail. If this parameter is NULL, it MUST be ignored.

@WebImageWidth: An integer used by lists (1) to determine the rendering width of an image. If this parameter is NULL, it MUST be ignored.

@WebImageHeight: An integer used by lists (1) to determine the rendering height of an image. If this parameter is NULL, it MUST be ignored.

@bParentFolderChecked: A bit that specifies whether or not the list (1) parent folder (the folder in which the list (1) is contained) has been checked for existence.

@OnRestore: A **Boolean** value indicating that this list (1) is undergoing a back-up restore operation.

@EventSinkAssembly: An **assembly name** for an **event handler** of an **event sink** for the list (1). If this parameter is NULL, it MUST be ignored.

@EventSinkClass: An **assembly class identifier (CLSID)** for an event handler of an event sink for the list (1). If this parameter is NULL, it MUST be ignored.

@EventSinkData: Event sink data for an event handler of an event sink for the list (1). If this parameter is NULL, it MUST be ignored.

@ContentTypes: The XSD representation of the content types available to this list (1).

@RootFolderId: The identifier (a GUID) of the **root folder** for this list (1).

@TitleResource: An optional Unicode string used to determine the list (1) title that is appropriate for a **multilingual user interface (MUI)**. If this parameter is not NULL, it MUST be a **resource token** or the name of a user resource.

@DescriptionResource: An optional Unicode string used to determine the list (1) description that is appropriate for a MUI. If this parameter is not NULL, it MUST be a resource token or the name of a user resource.

@ValidationFormula: An optional Unicode string that can be used to perform custom validation rules prior to the list (1) being updated. If specified this MUST be an expression specified by [\[MS-WSSTS\]](#) section 2.9.

@ValidationMessage: An optional Unicode string, or strings, that are suitable to display in a user interface when the list (1) fails validation based on **@ValidationFormula**.

@FolderFullUrlRet: The full URL of the list (1).

@TimeCreated: The date and time the list (1) was created.

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

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

Value	Description
0	Successful execution.
3	The system cannot find the URL specified.
5	Access is denied.
80	The list (1) with the specified title already exists.
87	Invalid parameter specified (@SiteId , @WebId , @FolderNameBase).
212	The path segment is locked and cannot be reallocated.
1836	Not enough quota is available to process this command.

Result sets: The stored procedure MUST return 1 result set when the return code is zero ("0"). Otherwise, it MUST not return any result sets.

3.1.4.19.1 List Metadata Result Set

Returns the list (1) metadata for the newly created list (1). This result set will be returned when input parameter **@BaseType** has a value of 1. This result set is defined in the [\[MS-WSSFO2\]](#), section [2.2.5.12](#).

3.1.4.19.2 Id and Full URL Result Set

The **Id and Full URL Result Set** returns the list identifier and full URL of the new list (1). This result set will be returned when input parameter **@BaseType** has a value different than 1. The result set is defined using T-SQL syntax, as follows:

```
{ListId}          uniqueidentifier,  
{FolderFullUrl}  nvarchar(256);
```

{ListId}: Contains the list identifier of the list (1) that has been created.

{FolderFullUrl}: Contains the full URL of the list (1) that has been created.

3.1.4.20 proc_CreateSharedAccessRequest

The **proc_CreateSharedAccessRequest** stored procedure is called when a **co-authoring transition** request is made for a document.

```
PROCEDURE proc_CreateSharedAccessRequest ( @SiteId uniqueidentifier,  
      @DocDirName nvarchar(256),  
      @DocLeafName nvarchar(128),  
      @DocId uniqueidentifier OUTPUT,  
      @RequestGuid uniqueidentifier = NULL OUTPUT  
    );
```

@SiteId: The site collection identifier of the site collection which contains the document for which a co-authoring transition request was made.

@DocDirName: The directory name of the document.

@DocLeafName: The leaf name of the document.

@DocId: An output parameter containing the document identifier (2) of the document for which a co-authoring transition request was made.

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

Return Code Values: MUST return an integer value of zero on successful completion. The stored procedure MUST return a nonzero value if an error occurs.

Result Sets: MUST NOT return any result sets.

3.1.4.21 proc_CreateSite

The **proc_CreateSite** stored procedure is called to create a new site collection with the specified metadata. The stored procedure is defined by using T-SQL syntax, as follows:

```
PROCEDURE proc_CreateSite(  
    --
```



```

@SiteId                uniqueidentifier,
@DirName               nvarchar(256),
@LeafName              nvarchar(128),
@RootWebUrl            nvarchar(256),
@Language              int,
@Collation             smallint,
@CalendarType         smallint,
@Time24               bit,
@OwnerSID              varbinary(512),
@OwnerLogin            nvarchar(255),
@OwnerName             nvarchar(255),
@OwnerEmail           nvarchar(255),
@SecondaryContactSID  varbinary(512),
@SecondaryContactLogin nvarchar(255),
@SecondaryContactName nvarchar(255),
@SecondaryContactEmail nvarchar(255),
@AdminsName            nvarchar(255),
@AdminsDescription    nvarchar(512),
@AdminsPermMask       bigint,
@AuthorsName          nvarchar(255),
@AuthorsDescription   nvarchar(512),
@AuthorsPermMask      bigint,
@ContributorsName     nvarchar(255),
@ContributorsDescription nvarchar(512),
@ContributorsPermMask bigint,
@BrowsersName         nvarchar(255),
@BrowsersDescription  nvarchar(512),
@BrowsersPermMask     bigint,
@GuestsName           nvarchar(255),
@GuestsDescription    nvarchar(512),
@GuestsPermMask       bigint,
@SiteHashKey          binary(16),
@HostHeader           nvarchar(260),
@UIVersion            tinyint,
@RequestGuid          uniqueidentifier = NULL OUTPUT
);

```

@SiteId: The site collection identifier of the site collection to be created.

@DirName: The directory name of the specified location. This parameter **MUST** be empty in the case of the **host header** site collection.

@LeafName: The leaf name of a site collection. This parameter **MUST** be empty in the case of the host header site collection.

@RootWebUrl: The virtual path relative to the **top-level site**. This parameter **MUST** be empty in the case of the host header site collection.

@Language: The LCID for the new site collection.

@Collation: The identifier that specifies the collation order.

@CalendarType: The identifier that specifies the **calendar type** that is being used.

@Time24: Specifies whether a 24-hour time format **SHOULD** be used when displaying time values. If this parameter is set to 1, the 24-hour time format **SHOULD** be used; otherwise, the 12-hour time format **SHOULD** be used.

@OwnerSID: The **SystemID** of the owner of the site collection.

@OwnerLogin: The login name of the owner of the site collection.

@OwnerName: The **display name** of the owner of the site collection.

@OwnerEmail: The e-mail address of the owner of the site collection.

@SecondaryContactSID: The SystemID of the secondary contact of the site collection.

@SecondaryContactLogin: The login name of the secondary contact of the site collection. This parameter is ignored when **@SecondaryContactSID** is NULL or equal to **@OwnerSID**.

@SecondaryContactName: The display name of the secondary contact of the site collection. This parameter is ignored when **@SecondaryContactSID** is NULL or equal to **@OwnerSID**.

@SecondaryContactEmail: The e-mail address of the secondary contact of the site collection. This parameter is ignored when **@SecondaryContactSID** is NULL or equal to **@OwnerSID**.

@AdminsName: The display name of the site group for administrators.

@AdminsDescription: The description for the site group for administrators.

@AdminsPermMask: An **access mask** containing the rights that MUST be granted to the site group for administrators in the site collection.

@AuthorsName: The display name of the site group for site authors.

@AuthorsDescription: The description of the site group for site authors.

@AuthorsPermMask: An access mask containing the rights that MUST be granted to the site group for site authors.

@ContributorsName: The display name of the site group for site contributors.

@ContributorsDescription: The description of the site group for site contributors.

@ContributorsPermMask: An access mask containing the rights that MUST be granted to the site group for site contributors.

@BrowsersName: The display name of the site group for site browsers.

@BrowsersDescription: The description of the site group for site browsers.

@BrowsersPermMask: An access mask containing the list of rights that MUST be granted to the site group for site browsers.

@GuestsName: The display name of the site group for site guests.

@GuestsDescription: The description of the site group for site guests.

@GuestsPermMask: An access mask containing the rights that MUST be granted to the site group for site guests.

@SiteHashKey: The hash key of this site collection. It is a random set of 16 bytes which are used to generate the **form digest validation** for this site collection. This parameter can be NULL.

@HostHeader: The host header of this site collection. This parameter MUST be NULL when it is not a host header.

@UIVersion: A user interface (UI) version number that is associated with the newly created site collection. If this parameter is set to NULL, 3 will be used as the default value.

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

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

Value	Description
0	Successful execution.
80	The site collection identifier already exists in the database.

Result sets: MUST return up to seven result sets. Some of the result sets are returned conditionally. Result sets from this stored procedure SHOULD be ignored by the caller. All result sets returned will be sent in the order specified in the following sections.

3.1.4.21.1 Site Owner Audit Mask Result Set

The **Site Owner Audit Mask Result Set** returns the information about the **audit flags** associated with the site (2) owner. The **Site Owner Audit Mask Result Set** MUST return a single row. It is defined in [\[MS-WSSEUX2\]](#) section 3.1.5.52.1.

3.1.4.21.2 Site Secondary Contact Audit Mask Result Set

The **Site Secondary Contact Site Audit Mask Result Set** returns the information about the Audit Flags associated with the site (2) secondary contact. This result set MUST NOT be returned if the **@SecondaryContactSID** parameter is NULL. For information about the definition, refer to [\[MS-WSSEUX2\]](#), section [3.1.5.52.1](#).

3.1.4.21.3 Site Administrator Audit Mask Result Set

The **Site Administrator Audit Mask Result Set** returns the information about the Audit Flags associated with the site (2) administrator. For information about the definition, refer to [\[MS-WSSEUX2\]](#), section [3.1.5.52.1](#).

3.1.4.21.4 Site Author Audit Mask Result Set

The **Site Author Audit Mask Result Set** returns the information about the Audit Flags associated with the site (2) author. For information about the definition, refer [\[MS-WSSEUX2\]](#), section [3.1.5.52.1](#).

3.1.4.21.5 Site Contributor Audit Mask Result Set

The **Site Contributor Audit Mask Result Set** returns the information about the Audit Flags associated with the site (2) contributor. For information about the definition, refer to [\[MS-WSSEUX2\]](#), section [3.1.5.52.1](#).

3.1.4.21.6 Site Browser Audit Mask Result Set

The **Site Browser Audit Mask Result Set** returns the information about the Audit Flags associated with the site (2) browser. For information about the definition, refer to [\[MS-WSSEUX2\]](#), section [3.1.5.52.1](#).

3.1.4.21.7 Site Guest Audit Mask Result Set

The **Site Guest Audit Mask Result Set** returns the information about the Audit Flags associated with the site (2) guest. For information about the definition, refer to [\[MS-WSSEUX2\]](#), section [3.1.5.52.1](#).

3.1.4.22 proc_CreateView

The **proc_CreateView** stored procedure is called to create a new view for the specified list (1). The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_CreateView(
    @SiteId                uniqueidentifier,
    @WebId                 uniqueidentifier,
    @ViewId                uniqueidentifier,
    @Level                 tinyint,
    @ListId                uniqueidentifier,
    @Type                  tinyint,
    @Flags                 int,
    @BaseViewID            tinyint,
    @DisplayName            nvarchar(255),
    @ContentTypeId         varbinary(512),
    @DocId                 uniqueidentifier,
    @WebPartTypeId         uniqueidentifier,
    @ZoneId                 nvarchar(64),
    @PartOrder             int,
    @ViewOrder             int,
    @View                  varbinary(max) = NULL,
    @source                 nvarchar(max) = NULL,
    @RequestGuid           uniqueidentifier = NULL OUTPUT
);
```

@SiteId: The site collection identifier of the site collection to contain the newly created view for the specified list (1).

@WebId: The site identifier of the site (2) which contains the newly created view for the specified list (1).

@ViewId: A **view identifier** for the newly created view for the specified list (1).

@Level: The publishing level for the newly created view. This parameter MUST be 1.

@ListId: The list identifier of the list (1) for which the new view is being created.

@Type: The **page type** for the newly created view. See [\[MS-WSSFO2\]](#), section [2.2.3.14](#) for valid values.

@Flags: The **view flags** for the newly created view. See [\[MS-WSSFO2\]](#), section [2.2.2.12](#) for valid values.

@BaseViewID: The **base view identifier** for the newly created view. MUST be unique per view defined for the specified list (1).

@DisplayName: The Unicode string which represents a name for the newly created view.

@ContentTypeId: The content type identifier for the newly created view.

@DocId: The document identifier (2) of the document that will contain the newly created view. This parameter MUST NOT be NULL.

@WebPartTypeId: The **Web Part type identifier** of the Web Part for the newly created view. This parameter MUST NOT be NULL.

@ZoneId: The name of the **Web Part zone identifier** that the specified Web Part will be positioned in the specified document.

@PartOrder: A **zero-based index** integer specifying the order that the Web Part will be displayed in the specified document. Each Web Part within the specified **Web Part zone** will be displayed in increasing order. MUST NOT be NULL.

@ViewOrder: A zero-based index integer specifying the order that the newly created view will be shown in relation to existing views for the specified list (1). When displaying views for a given list (1), the views MUST be listed in increasing order as specified by **@ViewOrder**. If this parameter is NULL, then the newly created view MUST be assigned an order greater than the order of all other views for the specified list (1).

@View: A compressed structure. See [MS-WSSFO2], section [2.2.4.8](#) for valid values. Uncompressed it is expressed in **Collaborative Application Markup Language (CAML)** used when processing this view. See [\[MS-WSSCAML\]](#) for more information about Collaborative Application Markup Language (CAML). The query is compressed by the algorithm specified in [\[RFC1950\]](#).

@source: The **Web Part property** or properties of the Web Part in either wpv2:WebPart format (see [\[MS-WPPS\]](#), section [2.2.3.2](#) for more information regarding the wpv2:WebPart format) or HTML format.

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

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

Value	Description
0	The stored procedure execution has finished. The stored procedure might have failed.
1816	The view for the specified list (1) could not be created because creating the new view would have exceeded the site collection quota.
212	The view for the specified list (1) could not be created because the site collection has the 0x00000001 bit set for its site collection flag. For more information regarding the site collection flags, see [MS-WSSFO2], section 2.2.2.9 .
3	The document specified by the @DocId stored procedure parameter exists in a site (2) whose site identifier is not the same as the site identifier specified by the @WebId stored procedure parameter.
1	An error occurred and the view was not created.

Result sets: MUST NOT return any result sets.

3.1.4.23 proc_CreateWeb

The **proc_CreateWeb** stored procedure is called to create a new site (2). The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_CreateWeb (
```

```

@WebSiteId      uniqueidentifier,
@WebId          uniqueidentifier,
@WebDirName     nvarchar(256),
@WebLeafName    nvarchar(128),
@WebFullUrl     nvarchar(260),
@ProductVersion smallint,
@TemplateVersion smallint,
@Language       int,
@Collation      smallint,
@CalendarType   smallint,
@AuthorID       int,
@Time24         bit,
@ConvertIfThere bit,
@UniqueWeb      bit,
@UIVersion      tinyint,
@NewWebId       uniqueidentifier = NULL,
@DocId          uniqueidentifier = NULL,
@RequestGuid    uniqueidentifier = NULL OUTPUT
);

```

@WebSiteId: The site collection identifier of the site collection that will contain the new site (2).

@WebId: The site identifier of an existing site (2) that will become the parent site for the site (2) to be created.

@WebDirName: The directory name for the new site (2).

@WebLeafName: The leaf name for the new site (2).

@WebFullUrl: MUST be NULL and MUST be ignored by the server.

@ProductVersion: MUST be set to 4.

@TemplateVersion: The **site definition version** of the **site definition** for the new site (2).

@Language: The LCID for the new site (2).

@Collation: The collation order for the new site (2).

@CalendarType: The calendar type for the new site (2).

@AuthorID: The user identifier of the user that is creating the new site (2).

@Time24: MUST be one of the following:

Value	Description
0	If the new site (2) is to display time using the 12-hour clock notation .
1	If the new site (2) is to display time using the 24-hour clock notation .

@ConvertIfThere: If this parameter is set to 1 and an existing folder is located at the URL specified by combining the **@WebDirName** and **@WebLeafName** parameters, then the existing folder is converted to a site (2). If this parameter is set to zero and an existing folder is located at the URL specified by combining the **@WebDirName** and **@WebLeafName** parameters, then the existing folder is not converted to a site (2).

@UniqueWeb: When the new site (2) is created, the role assignments from the site collection are applied to (or inherited by) the site (2). Therefore, the security scope of the various role assignments applied to the site collection also applies to any site (2) contained within the site collection. The site collection and its site (2) are said to have the same security scope. However, if this parameter is set to 1, the new site (2) will still inherit the role assignments of its site collection, but the site (2) will no longer have the same security scope as its site collection.

@UIVersion: A user interface (UI) version number that is associated with the newly created site (2). If this parameter is set to NULL, then 3 will be used as the default value.

@NewWebId: MUST be NULL and MUST be ignored by the server.

@DocId: MUST be NULL and MUST be ignored by the server.

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

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

Value	Description
0	Successful execution.
3	This return code MUST be returned if any of the following conditions apply: <ul style="list-style-type: none"> ▪ The site collection specified by the @WebSiteId parameter does not exist. ▪ The site (2) specified by the @WebId parameter exists, but is not contained within the site collection specified by the @WebSiteId parameter. ▪ The site (2) specified by the @WebId parameter does not exist.
80	This return code MUST be returned if any of the following conditions apply: <ul style="list-style-type: none"> ▪ A document or site (2) already exists at the URL specified by combining the @WebDirName and @WebLeafName parameters. ▪ A failure occurred and the new site (2) might not have been created.
85	If the URL specified by combining the @WebDirName and @WebLeafName parameters is an existing folder and the @ConvertIfThere parameter is zero, the new site (2) is created but will not be operational because the folder could not be converted to a functional site (2).
161	The site (2) creation operation MUST fail with this return code if all of the following conditions apply: <ol style="list-style-type: none"> 1. The URL specified by combining the @WebDirName and @WebLeafName parameters is an existing folder. 2. The @ConvertIfThere parameter is 1. 3. The existing folder or any of its child folders is already contained within another site (2).
138	The site (2) creation operation MUST fail with this return code if all of the following conditions apply: <ol style="list-style-type: none"> 1. The URL specified by combining the @WebDirName and @WebLeafName

Value	Description
	<p>parameters is an existing folder.</p> <ol style="list-style-type: none"> The @ConvertIfThere parameter is 1. The existing folder or any of its child folders is already contained within a list.
33	<p>The site (2) creation operation MUST fail with this return code if all of the following conditions apply:</p> <ol style="list-style-type: none"> The URL specified by combining the @WebDirName and @WebLeafName parameters is an existing folder. The @ConvertIfThere parameter is 1. The existing folder or any of its child folders contains a document that is checked out.
206	<p>The site (2) creation operation MUST fail with this return code if all of the following conditions apply:</p> <ol style="list-style-type: none"> The URL specified by combining the @WebDirName and @WebLeafName parameters is an existing folder. The @ConvertIfThere parameter is 1. The existing folder or any of its child folders exceeds the maximum directory name of 256 Unicode characters.
212	<p>The site (2) could not be created because the site collection has its 0x00000001 bit site collection flag set. See [MS-WSSFO2], section 2.2.2.9.</p>
1816	<p>The site (2) could not be created because creating the new site (2) would have exceeded the site collection quota.</p>

Result sets: MUST return the **Audit Flags Result Set** (section [3.1.4.23.1](#)) if the **@UniqueWeb** parameter is set to 1 on input. If the **@UniqueWeb** parameter is set to something other than 1 on input, **proc_CreateWeb** MUST NOT return the **Audit Flags Result Set**.

3.1.4.23.1 Audit Flags Result Set

The **proc_CreateWeb** stored procedure (section [3.1.4.23](#)) returns the audit flags (see [\[MS-WSSFO2\]](#), section [2.2.2.1](#) for additional information about audit flags) for the newly created site (2). The Audit Flags result set MUST return 1 row. The **Audit Flags Result Set** is defined using T-SQL syntax, as follows:

```
{WebId}                uniqueidentifier,
{WebAuditFlags}        int,
{WebInheritAuditFlags} int,
{SiteCollectionAuditFlags} int;
```

{WebId}: The site identifier for the newly created site (2).

{WebAuditFlags}: The audit flags (see [MS-WSSFO2], section [2.2.2.1](#) for additional information about audit flags) for the newly created site (2).

{WebInheritAuditFlags}: The audit flags (see [MS-WSSFO2], section [2.2.2.1](#) for additional information about audit flags) that are inherited from the parent site of the newly created site (2).

{SiteCollectionAuditFlags}: The audit flags (see [MS-WSSFO2], section [2.2.2.1](#) for additional information about audit flags) of the site collection that contains the newly created site (2).

3.1.4.24 proc_DeleteAllItemVersions

The **proc_DeleteAllItemVersions** stored procedure is called to delete all historical versions of a given list item. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_DeleteAllItemVersions(  
    @SiteId                uniqueidentifier,  
    @WebId                 uniqueidentifier,  
    @ListId               uniqueidentifier,  
    @ItemId               int,  
    @UserId               int,  
    @UseNvarchar1ItemName bit = 1,  
    @DeleteOp             int = 3,  
    @RequestGuid          uniqueidentifier = NULL OUTPUT  
);
```

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

@WebId: A site identifier of the site (2) containing the specified list item.

@ListId: A list identifier for the list (1) containing the specified list item.

@ItemId: The item identifier for the specified list item.

@UserId: The user identifier for the current user.

@UseNvarchar1ItemName: A bit flag specifying whether to use the nvarchar1 column value of the **AllUserData** table for the list item's display name. For more information regarding the nvarchar1 column of the **AllUserData** table, see [\[MS-WSSFO2\]](#) section 2.2.7.3.

@DeleteOp: A parameter specifying the delete options. The value MUST be listed in the following table.

Value	Description
3	The deleted document versions MUST NOT be placed in the Recycle Bin (a delete operation that cannot be recovered).
4	The deleted document versions MUST be placed in the Recycle Bin (recoverable delete).

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

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

Value	Description
0	Successful execution.

Value	Description
2	The list item specified cannot be found in the site collection.

Result sets: MUST NOT return any result sets.

3.1.4.25 `proc_DeleteAttachment`

The **`proc_DeleteAttachment`** stored procedure is called to delete an attachment from a list item. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_DeleteAttachment (
    @SiteId                uniqueidentifier,
    @WebId                 uniqueidentifier,
    @FolderUrl             nvarchar(256),
    @RowID                 uniqueidentifier,
    @UserId                int,
    @DeleteOp              int,
    @RequestGuid           uniqueidentifier = NULL OUTPUT
);

```

@SiteId: The site collection identifier of the site collection which contains the specified site (2).

@WebId: The site identifier of the site (2) which contains the specified **list item attachment** to be deleted.

@FolderUrl: The **store-relative URL** path of the list item containing the attachment to be deleted.

@RowID: The **attachment identifier** for the attachment to be deleted.

@UserId: The identifier for the user performing the delete operation.

@DeleteOp: The value that specifies the type of delete operation to perform. This parameter MUST specify a valid value from the following table.

Value	Description
3	Delete attachment.
4	Send attachment to the Recycle Bin. If Recycle Bin is not available, do not delete the attachment.
5	Send attachment to the Recycle Bin. If Recycle Bin is not available, delete the attachment.

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

Return values: An integer that MUST be zero.

Result sets: MUST NOT return any result sets.

3.1.4.26 `proc_DeleteAttachmentsFolder`

The **`proc_DeleteAttachmentsFolder`** stored procedure is called to disable attachments on a list (1) and remove all existing attachments of that list. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_DeleteAttachmentsFolder(
    @SiteId          uniqueidentifier,
    @WebId           uniqueidentifier,
    @ListId          uniqueidentifier,
    @UserId          int
);

```

@SiteId: The site collection identifier of the site collection which contains the specified site (2).

@WebId: The site identifier of the site (2) which contains the specified list (1).

@ListId: The list identifier of the list (1) which contains the attachments to be deleted.

@UserId: The identifier for the user performing the delete operation.

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

Value	Description
0	Successful execution.
5	The user specified by @UserId is not authorized to make this change.
33	Cannot delete attachments folder containing checked out or locked files.
50	Cannot delete attachments folder because the folder's URL is malformed.
1150	Concurrency violation or unknown error occurred.

Result sets: MUST NOT return any result sets.

3.1.4.27 proc_DeleteChanges

The **proc_DeleteChanges** stored procedure is called to delete changes from the change log and alert (1) events from the alert event log older than the specified number of days. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_DeleteChanges(
    @DaysKeepChanges    int,
    @DaysKeepEvents     int,
    @ChangesDeleted     int = 0 OUTPUT,
    @EventsDeleted      int = 0 OUTPUT,
    @RequestGuid        uniqueidentifier = NULL OUTPUT
);

```

@DaysKeepChanges: An integer that specifies a number of days from the present date to keep changes in the change log. Changes in the change log older than the number of days specified by **@DaysKeepChanges** will be deleted. A value of -1 indicates that all changes MUST be deleted. A negative value other than -1 MUST NOT be specified.

@DaysKeepEvents: An integer that specifies a number of days from the present date to keep alert (1) events in the alert event log. Alert events in the alert event log older than the number of days specified by **@DaysKeepEvents** will be deleted. A value of -1 indicates that all alert (1) events MUST be deleted. A negative value other than -1 MUST NOT be specified.

@ChangesDeleted: An output parameter containing the number of changes deleted from the change log.

@EventsDeleted: An output parameter containing the number of alert (1) events deleted from the alert job queue.

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

Return values: An integer that MUST be zero.

Result sets: MUST NOT return any result sets.

3.1.4.28 `proc_DeleteEventLog`

The **`proc_DeleteEventLog`** stored procedure is called to delete both alert subscriptions from the alert subscription list, and alert (1) events from the alert event log, respectively. Alert subscriptions and alert (1) events older than the specified Coordinated Universal Time (UTC) time will be deleted. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_DeleteEventLog(  
    @SiteId                uniqueidentifier,  
    @EventTime             datetime,  
    @RequestGuid           uniqueidentifier = NULL OUTPUT  
);
```

@SiteId: The site collection identifier of the site collection for which the alert subscriptions and alert (1) events will be deleted.

@EventTime: A Coordinated Universal Time (UTC) time. Alert events created before this time will be deleted.

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

Return Code Values: An integer that MUST be zero.

Result Sets: MUST NOT return any result sets.

3.1.4.29 `proc_DeleteFileFragmentsByTag`

The **`proc_DeleteFileFragmentsByTag`** stored procedure is called to delete the file fragments for a document based on values of a file fragment tag and a file fragment identifier. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_DeleteFileFragmentsByTag (  
    @SiteId                uniqueidentifier,  
    @DocId                 uniqueidentifier,  
    @DocVersion            int,  
    @ContentVersion        int,  
    @UserId                int,  
    @Partition             tinyint,  
    @Tag                   varbinary(40),  
    @Id                    bigint,  
    @DeleteUntil           bit,  
    @CheckPerms            bit,  
    @UpdateQuota           bit,  
    @QuotaChange           bigint OUTPUT
```

);

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

@DocId: The document identifier (2) of the document associated with the file fragments being deleted.

@DocVersion: The current internal version number of the document. This parameter MUST be ignored if **@CheckPerms** is not 1.

@ContentVersion: The current content version number of the document to update. This parameter MUST be ignored if **@CheckPerms** is not 1.

@UserId: The user identifier of the current user making the request. This parameter MUST be ignored if **@CheckPerms** is not 1.

@Partition: The Identifier for a file fragment partition of the file fragment partition where the file fragments to delete are located.

@Tag: The file fragment tag of the file fragments to delete.

@Id: This parameter MUST be ignored if **@DeleteUntil** is not 1.

@DeleteUntil: If this parameter is 1, file fragments with file fragment identifier less than @Id MUST be deleted. Else all file fragments MUST be deleted.

@CheckPerms: If this parameter is 1, **proc_FileFragmentPermissionCheck** MUST be called with parameters **@SiteId**, **@DocId**, **@UserId**, **@DocVersion**, and **@ContentVersion**.

@UpdateQuota: If this parameter is 1, the size in bytes of the file fragments deleted MUST be reflected in the quota.

@QuotaChange: The net amount of change to the file fragment data deleted, in bytes. This parameter MUST not be set to NULL.

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

Value	Description
0	Successful execution.
212	@UpdateQuota is 1 and the site collection is over quota or is write-locked.
4317	There were no file fragments deleted.
	Return codes from proc_FileFragmentPermissionCheck can be returned here.

Result sets: MUST NOT return any result sets.

3.1.4.30 **proc_DeleteItemVersion**

The **proc_DeleteItemVersion** stored procedure is called to delete a historical version of a list item. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_DeleteItemVersion(  
    @SiteId                uniqueidentifier,
```

```

@WebId                uniqueidentifier,
@ListId               uniqueidentifier,
@ItemId               int,
@ItemVersion          int,
@UserId               int,
@UseNvarchar1ItemName bit = 1,
@DeleteOp             int = 3,
@RequestGuid          uniqueidentifier = NULL OUTPUT
);

```

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

@WebId: The site identifier for the site (2) containing the specified list item.

@ListId: The list identifier of the list (1) which contains the specified list item.

@ItemId: The item identifier for the specified list item.

@ItemVersion: The user interface (UI) version for the specified list item.

@UserId: The user identifier for the current user.

@UseNvarchar1ItemName: A bit flag specifying whether to use the nvarchar1 column value of the AllUserData table for the list item's display name. For more information regarding the nvarchar1 column of the AllUserData table, see [\[MS-WSSFO2\]](#) section 2.2.7.3.

@DeleteOp: A parameter specifying the delete options. The value MUST be listed in the following table.

Value	Description
3	The deleted document versions MUST NOT be placed in the Recycle Bin (non-recoverable delete).
4	The deleted document versions MUST be placed in the Recycle Bin (recoverable delete).

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

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

Value	Description
0	Successful execution.
2	The list item specified cannot be found in the site collection.

Result sets: MUST NOT return any result sets.

3.1.4.31 **proc_DeleteSite**

The **proc_DeleteSite** stored procedure is called to delete a site collection. The stored procedure is defined using T-SQL syntax, as follows:

```

PROCEDURE proc_DeleteSite(
@SiteId                uniqueidentifier,
@ThresholdRowCount     int = 0,
@RequestGuid           uniqueidentifier = NULL OUTPUT
)

```

);

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

@ThresholdRowCount: If this parameter is not set to NULL or zero, the site collection MUST NOT be deleted if the number of published list items for all lists (1) in the site collection exceeds this parameter value. However, if the parameter is set to NULL or zero, then an attempt will be made to delete the site. For more information regarding published list items, see [\[MS-WSSFO2\]](#), section [2.2.2.6](#).

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

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

Value	Description
0	The stored procedure execution has finished. The stored procedure might have failed.
36	The site collection was not deleted because the number of published list items for all lists (1) in the site collection exceeds the value specified by the @ThresholdRowCount parameter. If the @ThresholdRowCount parameter was set to NULL or zero, this return code MUST NOT be returned. For more information regarding published list items, see [MS-WSSFO2] , section 2.2.2.6 .

Result sets: MUST return two result sets as described in the following sections.

3.1.4.31.1 Site Collection Flags Result Set

The **Site Collection Flags Result Set** returns information about the site collection that has been deleted. The **Site Collection Flags Result Set** MUST return one row as defined in section [2.2.6.1](#).

3.1.4.31.2 Distribution List E-mail Address Result Set

The **Distribution List E-mail Address Result Set** returns information about the site collection that has been deleted as defined in section [2.2.6.2](#).

3.1.4.32 proc_DeleteSiteAsync

The **proc_DeleteSiteAsync** stored procedure is called to schedule a site collection for deletion. The stored procedure is defined using T-SQL syntax, as follows:

```
PROCEDURE proc_DeleteSiteAsync (  
    @SiteId          uniqueidentifier,  
    @Restorable      bit,  
    @RequestGuid     uniqueidentifier = null OUTPUT  
);
```

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

@Restorable: Specifies whether the action to schedule the site collection for deletion is reversible.

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

Return values: An integer that the protocol client MUST ignore.

Result sets: MUST return two result sets in the order of the following sections.

3.1.4.32.1 Site Collection Flags Result Set

The **Site Collection Flags Result Set** returns information about the site collection that has been deleted. The **Site Collection Flags Result Set** MUST return one row as defined in section [2.2.6.1](#)

3.1.4.32.2 Distribution List E-mail Address Result Set

The **Distribution List E-mail Address Result Set** returns information about the site collection that has been deleted as defined in Section [2.2.6.2](#)

3.1.4.33 proc_DeleteSiteCoreAsync

The **proc_DeleteSiteCoreAsync** stored procedure is called to delete all data associated with a site collection that is scheduled for deletion. The stored procedure is defined using T-SQL syntax, as follows:

```
PROCEDURE proc_DeleteSiteCoreAsync (  
    @SiteId          uniqueidentifier,  
    @DeletionId     bigint  
);
```

@SiteId: The site collection identifier of the site collection whose data has to be deleted.

@DeletionId: An integer that uniquely represents a site collection whose data is scheduled to be deleted. This MUST be the same value returned in the **Site Deletion Batch Result Set** (section [3.1.4.63.1](#)) for the corresponding value of **@SiteId**.

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

Value	Description
0	Successful execution.
2	The site collection specified by the @SiteId parameter is not scheduled for deletion.

Result sets: MUST NOT return any result sets.

3.1.4.34 proc_RestoreSite

The **proc_RestoreSite** stored procedure is called to remove a site collection from scheduled deletion. The stored procedure is defined using T-SQL syntax, as follows:

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

@SiteId: The site collection identifier of the site collection which is scheduled for deletion.

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

Value	Description
0	Successful execution.
2	The site collection specified by the @SiteId parameter is not scheduled for deletion, is being deleted, or is already deleted.

Result sets: MUST NOT return any result sets.

3.1.4.35 **proc_DeleteView**

The **proc_DeleteView** stored procedure is called to delete a view from the specified list (1). The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_DeleteView(
    @SiteId          uniqueidentifier,
    @ListId          uniqueidentifier,
    @ViewId         uniqueidentifier,
    @CanManagePersonalViews bit,
    @CanManageLists bit,
    @UserId         int,
    @RequestGuid    uniqueidentifier = NULL OUTPUT
);

```

@SiteId: The site collection identifier of the site collection that contains the specified view for the specified list (1).

@ListId: The list identifier of the list (1) that contains the specified view.

@ViewId: The view identifier of the view that is to be deleted.

@CanManagePersonalViews: If the specified view to be deleted is a personal view and this parameter is set to 1 and the user identifier specified by the **@UserId** parameter is the original creator of the view, then the view **MUST** be deleted. If the specified view to be deleted is a personal view and this parameter is set to zero, then the view **MUST NOT** be deleted.

@CanManageLists: If the specified view to be deleted is a **shared view** and this is set to 1, then the view **MUST** be deleted. If the specified view to be deleted is a shared view and this parameter is set to zero, then the view **MUST NOT** be deleted.

@UserId: The user identifier of the user that originally created the specified view. If the specified view to be deleted is a personal view and the user identifier specified by this parameter is the original creator of the view and the **@CanManagePersonalViews** parameter is set to 1, then the view **MUST** be deleted. If the specified view to be deleted is a personal view and the user identifier specified by this parameter is not the original creator of the view, then the view **MUST NOT** be deleted.

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

Value	Description
0	Successful execution.
3	The view specified by the @ViewId parameter does not exist or is not the current version.
5	This return code MUST be returned if any of the follow occur:

Value	Description
	<ul style="list-style-type: none"> ▪ The view specified by the @ViewId parameter already exists but is not a view for the list (1) specified by the @ListId parameter. ▪ The view specified by the @ViewId parameter is a personal view, and the @CanManagePersonalViews parameter is not 1. ▪ The view specified by the @ViewId parameter is a personal view, and the user identifier specified by the @UserId parameter is not the original creator of the specified view to be deleted. ▪ The view specified by the @ViewId parameter is a shared view, and the @CanManageLists parameter is not 1.

Result sets: MUST NOT return any result sets.

3.1.4.36 **proc_DeleteWeb**

The **proc_DeleteWeb** stored procedure is called to delete a site (2). This stored procedure MUST NOT be used to delete a top-level site. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_DeleteWeb (
    @WebSiteId                uniqueidentifier,
    @WebUrl                   nvarchar(260),
    @ThresholdRowCount        int = 0,
    @DeleteTransactionId      varbinary(16) = 0x,
    @FailedUrl                 nvarchar(260) = NULL OUTPUT,
    @DeleteFlags               int = 0,
    @WebIdDelete               uniqueidentifier = NULL,
    @RequestGuid               uniqueidentifier = NULL OUTPUT
);

```

@WebSiteId: The site collection identifier of the site collection that contains the specified site (2) to be deleted.

@WebUrl: The URL of the site (2) in store-relative form to be deleted.

@ThresholdRowCount: If this parameter is not set to NULL or zero, the site (2) MUST NOT be deleted if the number of published list items for all lists (1) in the site (2) exceeds the value of this parameter. Otherwise, if this parameter is set to NULL or zero, it is ignored. For more information about published list items, see [\[MS-WSSFO2\]](#), section [2.2.2.6](#).

@DeleteTransactionId: The delete transaction identifier of the Recycle Bin item corresponding to the site (2) being deleted. This parameter MUST be 0x.

@FailedUrl: If this parameter is not NULL and there is a failure when deleting the site (2), this parameter MAY be assigned the URL in store-relative form of the document that failed to be deleted from the site (2).

@DeleteFlags: The **delete flags** is used to perform additional operations for the site (2) to be deleted. MUST be one of the delete flags defined in section [2.2.3.4](#). If this is zero, then it is ignored and the site (2) is deleted.

@WebIdDelete: If this parameter is not NULL, then this is the site identifier of the site (2) that will be deleted if a site (2) does not exist at the URL specified by the **@WebUrl** parameter.

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

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

Value	Description
0	The site (2) might have been deleted. If a site collection specified by the @WebSiteId parameter does not exist, the site (2) MUST not be deleted. If a site collection specified by the @WebSiteId parameter does not exist, then the site (2) MUST not be deleted.
3	This return code is returned if the @DeleteFlags parameter is set to 8 and a site (2) does not exist at the URL specified by the @WebUrl parameter in the site collection specified by the @WebSiteId parameter. This return code is also returned if the @WebUrl is null.
5	This return code is returned if the site (2) exists at the URL specified by the @WebUrl parameter, but it does not have a parent site and the @DeleteFlags parameter did not specify 8 as its delete flags.
33	The site (2) deletion operation MUST fail with this return code if all of the following are true: <ul style="list-style-type: none">▪ The URL specified by the @WebUrl parameter is an existing folder.▪ The @DeleteFlags parameter did not specify 8 as its delete flags.▪ The existing folder or any of its child folders contain documents that are checked out.
36	The site (2) was not deleted because the number of published list items for all lists (1) in the site (2) exceeds the value specified by the @ThresholdRowCount parameter. If the @ThresholdRowCount parameter was set to NULL or zero, this return code MUST NOT be returned. For more information regarding published list items, see [MS-WSSFO2], section 2.2.2.6 .
161	The site (2) deletion operation MUST fail with this return code if the following is true: <ul style="list-style-type: none">▪ The URL specified by the @WebUrl parameter is an existing folder.▪ The @DeleteFlags parameter did not specify 8 as its delete flags.
138	The site (2) deletion operation MUST fail with this return code if all of the following are true: <ul style="list-style-type: none">▪ The URL specified by the @WebUrl parameter is an existing folder.▪ The @DeleteFlags parameter did not specify 8 as its delete flags.▪ The existing folder or any of its child folders is already contained within a list (1).
206	The site (2) deletion operation MUST fail with this return code if the following is true: <ul style="list-style-type: none">▪ The URL specified by the @WebUrl parameter is an existing folder.▪ The @DeleteFlags parameter did not specify 8 as its delete flags.▪ The existing folder or any of its child folders exceeds the maximum directory name of 256 Unicode characters.

Result sets: MUST return the **Audit Flags Result Set** (section [3.1.4.36.1](#)) if the site (2) was successfully deleted.

3.1.4.36.1 Audit Flags Result Set

The **proc_DeleteWeb** stored procedure (section [3.1.4.36](#)) returns the audit flags for the deleted site (2) and for the site collection that contains the site (2). The **Audit Flags Result Set** MUST return one row. For more information regarding the audit flags, see [\[MS-WSSFO2\]](#), section [2.2.2.1](#). The T-SQL syntax for the result set is as follows:

```
{WebId}                uniqueidentifier,  
{WebAuditFlags}        int,  
{WebInheritAuditFlags} int,  
{SiteAuditFlags}       int;
```

{WebId}: The site identifier of the site (2) that was deleted.

{WebAuditFlags}: The audit flags of the site (2) that was deleted. See [\[MS-WSSFO2\]](#), section [2.2.2.1](#) for additional information regarding audit flags.

{WebInheritAuditFlags}: The audit flags that are inherited from the parent site of the deleted site (2). See [\[MS-WSSFO2\]](#), section [2.2.2.1](#) for additional information regarding audit flags.

{SiteAuditFlags}: The audit flags of the site collection that contains the deleted site (2). See [\[MS-WSSFO2\]](#), section [2.2.2.1](#) for additional information regarding audit flags.

3.1.4.37 proc_RecycleWeb

The **proc_RecycleWeb** stored procedure is called to move a site (2) into the Recycle Bin. This stored procedure MUST NOT be used to move a top-level site into the Recycle Bin. The T-SQL syntax for the stored procedure is as follows.

```
PROCEDURE proc_RecycleWeb (  
    @SiteId                uniqueidentifier,  
    @WebUrl                 nvarchar(260),  
    @UserId                 int,  
    @ThresholdRowCount      int = 0,  
    @FailedUrl              nvarchar(260) = NULL OUTPUT,  
    @RequestGuid            uniqueidentifier = NULL OUTPUT  
);
```

@SiteId: The site collection identifier of the site collection that contains the specified site (2) to delete.

@WebUrl: The URL of the site (2) in store-relative form to delete.

@UserId: The identifier of the current user performing the operation. This value MUST NOT be NULL.

@ThresholdRowCount: If this parameter is not set to NULL or zero, the site (2) MUST NOT be deleted if the number of published list items for all lists (1) in the site (2) exceeds the value of this parameter. Otherwise, if this parameter is set to NULL or zero, it is ignored. For more information about published list items, see [\[MS-WSSFO2\]](#), section [2.2.2.6](#).

@FailedUrl: If this parameter is not NULL and there is a failure when deleting the site (2), the parameter MAY be assigned the URL in store-relative form of the document that failed to be deleted from the site (2).

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

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

Value	Description
0	Successful execution of the operation.
3	This return code is returned if a site (2) does not exist at the URL specified by the @WebUrl parameter in the site collection specified by the @SiteId parameter.
5	This return code is returned if the site (2) exists at the URL specified by the @WebUrl parameter, but does not have a parent site.
33	The site (2) Recycle Bin operation MUST fail with this return code if all of the following are true: <ul style="list-style-type: none">▪ The URL specified by the @WebUrl parameter is an existing folder.▪ The existing folder or any of its child folders contain documents that are checked out.
36	The site (2) was not moved to the Recycle Bin because the number of published list items for all lists (1) in the site (2) exceeds the value specified by the @ThresholdRowCount parameter. If the @ThresholdRowCount parameter was set to NULL or zero, this return code MUST NOT be returned. For more information about published list items, see [MS-WSSFO2], section 2.2.2.6 .
161	The site (2) Recycle Bin operation MUST fail with this return code if the following is true: <ul style="list-style-type: none">▪ The URL specified by the @WebUrl parameter is an existing folder.
138	The site (2) Recycle Bin operation MUST fail with this return code if all of the following are true: <ul style="list-style-type: none">▪ The URL specified by the @WebUrl parameter is an existing folder.▪ The existing folder or any of its child folders is already contained within a list (1).
206	The site (2) Recycle Bin operation MUST fail with this return code if all of the following are true: <ul style="list-style-type: none">▪ The URL specified by the @WebUrl parameter is an existing folder.▪ The existing folder or any of its child folders exceeds the maximum directory name of 256 Unicode characters.

Result sets: MUST return the **Audit Flags Result Set** (section [3.1.4.37.1](#)) if the site (2) was successfully moved to the Recycle Bin.

3.1.4.37.1 Audit Flags Result Set

The **proc_RecycleWeb** stored procedure (section [3.1.4.37](#)) returns the audit flags for the site (2) that was moved to the Recycle Bin and for the site collection that contains the site (2). The **Audit Flags Result Set** MUST return one row. For more information regarding the audit flags, see [\[MS-WSSFO2\]](#), section [2.2.2.1](#). The T-SQL syntax for the result set is as follows:

```
{WebId}                uniqueidentifier,  
{WebAuditFlags}        int,  
{WebInheritAuditFlags} int,
```

```
{SiteAuditFlags} int;
```

{WebId}: The site identifier of the site (2) that was moved to the Recycle Bin.

{WebAuditFlags}: The audit flags of the site (2) that was moved to the Recycle Bin. See [MS-WSSFO2], section [2.2.2.1](#) for additional information regarding audit flags.

{WebInheritAuditFlags}: The audit flags that are inherited from the parent site of the site (2) that was moved to the Recycle Bin. See [MS-WSSFO2], section [2.2.2.1](#) for additional information regarding audit flags.

{SiteAuditFlags}: The audit flags of the site collection that contains the site (2) that was moved to the Recycle Bin. See [MS-WSSFO2], section [2.2.2.1](#) for additional information regarding audit flags.

3.1.4.38 `proc_DirtyDocWithForwardLinks`

The **`proc_DirtyDocWithForwardLinks`** stored procedure is called to mark all documents which have server relative forward links as dirty. The T-SQL syntax for the stored procedure is as follows:

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

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

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST NOT return any result sets.

3.1.4.39 `proc_DropListRecord`

The **`proc_DropListRecord`** stored procedure is called to delete a list item and optionally place the list item in the Recycle Bin. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_DropListRecord(
    @SiteId                uniqueidentifier,
    @WebId                 uniqueidentifier,
    @ListId                uniqueidentifier,
    @ServerTemplate        int,
    @Id                    int,
    @UseNvarchar1ItemName bit = 1,
    @AuditIfNecessary      bit = 0,
    @UserTitle             nvarchar(255),
    @Version               int = NULL,
    @UserId                int = 0,
    @NeedsAuthorRestriction bit = 0,
    @Basetype              int = NULL,
    @DeleteOp              int = 3,
    @eventData             varbinary(max) = NULL,
    @acl                   varbinary(max) = NULL,
    @ThresholdRowCount     int = 0,
    @DeleteTransactionId   varbinary(16) = 0x OUTPUT,
    @Size                  bigint = 0 OUTPUT,
    @IsCascadeDeleteOperation bit = 0,
    @IsCascadeParent       bit = 0,
```

```

    @ChildDeleteTransactionId    varbinary(16)    = NULL OUTPUT,
    @RequestGuid                 uniqueidentifier = NULL OUTPUT
);

```

@SiteId: The site collection identifier of the site collection which contains the specified list item to be deleted.

@WebId: The site identifier of the site (2) containing the list item to be deleted. This parameter MUST NOT be NULL.

@ListId: The list identifier of the list (1) containing the list item to be deleted.

@ServerTemplate: The integer value of the list template of the list (1) that contains the list item to be deleted.

@Id: The list item identifier of the list item to be deleted.

@UseNvarchar1ItemName: A bit flag specifying whether to use the nvarchar1 column value of the AllUserData table for the list item's display name. For more information regarding the nvarchar1 column of the AllUserData table, see [\[MS-WSSFO2\]](#) section 2.2.7.3.

@AuditIfNecessary: If this parameter is set to 1, the delete operation will be audited. Otherwise the operation will not be audited.

@UserTitle: This parameter MUST be set to NULL and MUST be ignored by the server.

@Version: An optional value to compare with the version number of the list item. It can be NULL. If this parameter is not NULL, the parameter MUST match the version number for successful completion.

@UserId: A identifier that is used for the current user. This value MUST NOT be NULL.

@NeedsAuthorRestriction: A bit flag specifying whether only the list item's author is permitted to delete the list item. It MUST NOT be NULL. If this parameter is set to 1, the current user specified by @UserId MUST be the list item's author for successful execution.

@Basetype: The base type of the list (1) containing the list item. This parameter SHOULD be ignored. The value MUST be listed in the following table.

Value	Description
0	Generic List
1	Document Library
3	Discussion Board
4	Survey List
5	Issues List

@DeleteOp: A parameter specifies the delete option. The value MUST be listed in the following table.

Value	Description
3	The deleted list item MUST NOT be placed in the Recycle Bin. Those list items that are not placed in the Recycle Bin cannot be recovered after the delete operation is successful.
4	The deleted list item MUST be placed in the Recycle Bin. Those list items that are placed in the Recycle Bin can be recovered after the delete operation is successful.

@eventData: Contains implementation-specific event data significant to the front-end Web server but otherwise opaque to the back-end database server to be stored by the back-end database server. It can be NULL.

@acl: The binary serialization of the WSS ACL Format for the data supplied in the **@eventData** parameter, to be stored with the data. It can be NULL. For more information regarding the WSS ACL Format, see [\[MS-WSSFO2\]](#) section 2.2.4.6.

@ThresholdRowCount: An optional parameter that if provided MUST be set to its default.

@DeleteTransactionId: A GUID which is the **transaction identifier** that encapsulated the actual delete operation. This is used so that multiple or hierarchical operations can be performed by the caller. If this is a zero length binary (0x) and @DeleteOp is set to 4, this procedure will define this as a new 16-byte identifier (converted GUID) and add a record to the Recycle Bin, otherwise it will use the passed in value in its work.

@Size: The size of the list item. It MUST NOT be NULL. This stored procedure MUST return the number of bytes used by the list item through the **@Size** parameter if the execution succeeded.

@IsCascadeDeleteOperation: An optional parameter that if provided, MUST be set to its default.

@IsCascadeParent: An optional parameter that if provided MUST be set to its default.

@ChildDeleteTransactionId: An optional parameter that if provided MUST be set to its default.

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

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

Value	Description
0	Successful execution.
3	The list item does not exist.
5	Access denied. The current user specified by @UserId parameter is not the same as the author of the list item when the @NeedsAuthorRestriction parameter is set to 1.
33	The attempt to delete directories that contain checked out files failed the operation.
1150	Concurrency violation. The @Version parameter does not match the version number of the list item. This value MUST only be returned if the @Version parameter is not NULL.

Result sets: MUST NOT return any result sets.

3.1.4.40 **proc_DropListUniqueField**

The **proc_DropListUniqueField** stored procedure is called to remove the **unique column constraint** for a field (1) in a list (1). The T-SQL syntax for the stored procedure is as follows:


```

PROCEDURE proc_DropListUniqueField (
    @SiteId          uniqueidentifier,
    @ListId          uniqueidentifier,
    @FieldId        uniqueidentifier
);

```

@SiteId: The site collection identifier of the site collection containing the specified list (1).

@ListId: The list identifier of the list (1) containing the specified field (1).

@FieldId: The field identifier of the specified field (1).

Return values: An integer that MUST be zero.

Result sets: MUST NOT return any result sets.

3.1.4.41 proc_DropLookupRelationship

The **proc_DropLookupRelationship** stored procedure is called to remove the existing relationship delete behavior from a relationship lookup field and converting the relationship lookup field to a lookup field. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_DropLookupRelationship (
    @SiteId          uniqueidentifier,
    @WebId          uniqueidentifier,
    @ListId          uniqueidentifier,
    @FieldId        uniqueidentifier,
    @TranLockStatus tinyint
);

```

@SiteId: The site collection identifier of the site collection which contains the specified list (1).

@WebId: The site identifier of the site (2) which contains the specified list (1).

@ListId: The list identifier of the list (1) containing the specified relationship lookup field.

@FieldId: The field identifier of the specified relationship lookup field.

@TranLockStatus: The value that specifies whether a short-term transaction application lock was taken for the specified site (2). This parameter MUST be a value from the following table.

Value	Description
3	The short-term transaction application lock was not taken for the specified site (2).
4	The short-term transaction application lock was taken for the specified site (2).

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

Value	Description
0	Successful execution.
301	The short-term transaction application lock was not taken for the specified site (2) and either a cascading behavior or restrict behavior is set on the relationship lookup field.

Result sets: MUST NOT return any result sets.

3.1.4.42 **proc_EnsureTranLockNotRequired**

The **proc_EnsureTranLockNotRequired** stored procedure is called to verify whether a short-term transaction application lock MUST be set on the site (2) containing a specified list (1). The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_EnsureTranLockNotRequired (  
    @SiteId                uniqueidentifier,  
    @ListId                uniqueidentifier,  
    @OnlyCheckForwardLink bit = 0  
);
```

@SiteId: The site collection identifier of the site collection containing the specified list (1).

@ListId: The list identifier of the specified list (1).

@OnlyCheckForwardLink: A bit flag specifying whether to only check if the specified list (1) has relationship lookup fields which have cascading behavior or restrict behavior.

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

Value	Description
0	The short-term transaction application lock MUST NOT be set on the site (2) containing the specified list (1) if all of the following conditions are true: <ul style="list-style-type: none">▪ The specified list (1) does not have any relationship lookup field which has cascading behavior or restrict behavior.▪ The bit flag specified by @OnlyCheckForwardLink is set to 1 or there is no relationship lookup field in any list (1) in the site (2) which has the list (1) specified by @ListId as the target list (1) and has cascading behavior or restrict behavior.
301	The short-term transaction application lock MUST be set on the site (2) containing the specified list (1) if either of the following conditions is true: <ul style="list-style-type: none">▪ The specified list (1) has at least one relationship lookup field which has cascading behavior or restrict behavior.▪ The bit flag specified by @OnlyCheckForwardLink is set to zero and there is at least one relationship lookup field in some list (1) in the site (2) which has the list (1) specified by @ListId as the target list (1) and has cascading behavior or restrict behavior.

Result sets: MUST NOT return any result sets.

3.1.4.43 **proc_FetchFileFormatMetaInfo**

The **proc_FetchFileFormatMetaInfo** stored procedure is called to read the file format metadata of a document. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_FetchFileFormatMetaInfo (  
    @DocId                uniqueidentifier,  
    @Level                tinyint,
```

```

        @DocVersion          int,
        @ContentVersion      int
    );

```

@DocId: The document identifier (2) of the document to read from.

@Level: The publishing level of the document to read from.

@DocVersion: The internal version number of the document to read from. If **@DocVersion** is not NULL and the current version number of the document is not equal to **@DocVersion**, the **FileFormatMetaInfo Rowset** MUST contain zero rows.

@ContentVersion: The content version of the document to read from. If **@ContentVersion** is not NULL and the current content version number of the document is not equal to **@ContentVersion**, the **FileFormatMetaInfo Rowset** MUST contain zero rows.

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

Value	Description
0	Successful execution.
21	The document with document identifier (2) equal to @DocId , publishing level equal to @Level , internal version equal to @DocVersion , and content version equal to @ContentVersion was not found.
30	There was an IO error reading the data.

Result wets: MUST return 1 **FileFormatMetaInfo Rowset**.

3.1.4.43.1 FileFormatMetaInfo Rowset

The **FileFormatMetaInfo Rowset** contains the file format metadata for a document. The T-SQL syntax for the result set is as follows:

```

FileFormatMetaInfo          varbinary(max);

```

FileFormatMetaInfo: The file format metadata for a document.

3.1.4.44 proc_FetchOldDoc

The **proc_FetchOldDoc** stored procedure is called to return a historical version of a document for the **HTTP GET** and **HTTP HEAD** operations for a specified user. Different sets of information are provided depending on the type of request (HTTP GET or HTTP HEAD). The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_FetchOldDoc(
    @DocSiteId          uniqueidentifier,
    @DocDirName         nvarchar(256),
    @DocLeafName        nvarchar(128),
    @IfModifiedSince    datetime,
    @FetchType          int,
    @ValidationType     int,
    @ClientVersion      int,

```

```

@ClientId                uniqueidentifier,
@SystemID                varbinary(512),
@VirusVendorID          int,
@ChunkSize              int,
@DGCacheVersion         bigint,
@PageView               tinyint,
@ThresholdRowcount      int,
@RequestGuid            uniqueidentifier = NULL OUTPUT
);

```

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

@DocDirName: The directory name of the document.

@DocLeafName: The leaf name of the document.

@IfModifiedSince: This parameter is used in combination with **@ValidationType** to determine whether the document stream will be returned. If the front-end Web server has a cached copy of the document stream, **@IfModifiedSince** MUST be the time in Coordinated Universal Time (UTC) the cached copy of the document was last modified. Otherwise, **@IfModifiedSince** MUST be NULL.

@FetchType: This parameter specifies the type of HTTP request. If set to zero, this specifies an HTTP GET request. If set to 1, this specifies an HTTP HEAD request. All values other than "1" MUST be treated as zero.

@ValidationType: This parameter is used to determine whether the document stream will be returned. It MUST be listed in the following table.

Value	Description
"0"	Return document stream.
"1"	Return document stream if @ClientId does not match the document identifier (2) of the document in the back-end database server.
"2"	Return document stream if @IfModifiedSince does not match the last modified time of the document in the back-end database server.
"3"	Return document stream if @IfModifiedSince does not match the last modified time of the document or @ClientId does not match the document identifier (2) of the document in the back-end database server.

@ClientVersion: The UI version of the requested document.

@ClientId: The document identifier (2) of the document used as an **HTTP entity tag** for client cache validation.

@SystemID: The SystemID of the user originating the request; NULL indicates an anonymous user.

@VirusVendorID: The identifier of the **virus scanner** registered for the **farm**.

@ChunkSize: Specifies the maximum size requested, in bytes, of the document. If the document size is larger than this maximum size, one the first **@ChunkSize** bytes MUST be returned for **{Content}** in the **Document Version Content Stream Result Set** (section [3.1.4.44.5](#)) and the front-end Web server can request the remainder of the document in a subsequent operation.

@DGCacheVersion: The version of the **domain group** cache as seen by the front-end Web server. It is used to compare with the domain group cache version in the back-end database server to determine whether an update is needed. A value of -2 is specified to indicate that the version numbers of the domain group cache are not requested.

@PageView: Non-NULL values indicate that the document is a list view webpage. This also indicates that information is needed to enable front-end Web server rendering based on metadata about the webpage, the item or items being rendered, the user's browsing context, the navigation structure of the site (2), and the user's security privileges. Information requested includes site (2) metadata, containing list (1) metadata, users and site groups metadata, webparts and related list (1) metadata. A NULL value indicates that the document is not a list view webpage.

@ThresholdRowCount: The maximum number of security scopes fetched by this stored procedure, which MUST be the same as the throttling limit.

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

Return values: `proc_FetchOldDoc` returns an integer return code which MUST be listed in the following table.

Value	Description
0	Successful execution.
3	The current version of the document was not found. Or The specified historical version of the document could not be found, or the requested version of the document is the current version of the document for the specified user.
18	Successful execution; based on values of @ValidationType, @ClientId, @ClientVersion, and @IfModifiedSince, the document was not fetched.

Result sets: MUST return the result sets in the following sections conditionally as described in each result set section.

3.1.4.44.1 Domain Group Cache Versions Result Set

The **Domain Group Cache Versions Result Set** returns the version numbers associated with the domain group cache for the site collection containing the document. The domain group cache contains a serialized binary representation of the **external groups** that are members (2) of the site groups.

The **Domain Group Cache Versions Result Set** MUST contain one row. If **@DGCacheVersion** is -2, then all columns returned will have the value -2 as well. The **Domain Group Cache Versions Result Set** is defined in [\[MS-WSSFO2\]](#) section 2.2.5.4.

3.1.4.44.2 Domain Group Cache Back-End Database Server Update Result Set

The **Domain Group Cache Back-End Database Server Update Result Set** contains information to be used in re-computing the domain group cache.

The **Domain Group Cache Back-End Database Server Update Result Set** returns only if **@DGCacheVersion** is not -2 and the real version of the domain group is more recent than the cached version (The value of **RealVersion** is greater than the value of **CachedVersion in the Domain Group Cache Versions Result Set**).

If the **Domain Group Cache Back-End Database Server Result Set** is returned, it indicates that the database's copy of the domain group cache is out of date and SHOULD be recomputed to ensure that proper security checks can be made. When returned, the **Domain Group Cache Back-End Database Server Update Result Set** MUST have one row. The **Domain Group Cache Back-End Database Server Update Result Set** is defined in [\[MS-WSSFO2\]](#) section 2.2.5.3.

3.1.4.44.3 Domain Group Cache Front-End Web Server Update Result Set

The **Domain Group Cache Front-End Web Server Update Result Set** contains the binary data needed to refresh the domain group cache in the front-end Web server.

The **Domain Group Cache Front-End Web Server Update Result Set** returns only if **@DGCacheVersion** is not -2 and the cached version is up to date (the value of **RealVersion** is not greater than the value of **CachedVersion** in the **Domain Group Cache Versions Result Set**).

The **Domain Group Cache Front-End Web Server Update Result Set** is defined in [\[MS-WSSFO2\]](#) section 2.2.5.5.

3.1.4.44.4 Document Version Metadata Result Set

The **Document Version Metadata Result Set** returns the document metadata for the specified version. If the current version of the document is not found, this result set MUST NOT be returned. If the specified version of the document is not found, this result set MUST be returned with zero rows. The T-SQL syntax for the result set is as follows:

```

Size                int,
DocFlags            int,
TimeCreated         datetime,
FullUrl             nvarchar(260),
{WebId}             uniqueidentifier,
FirstUniqueAncestorWebId uniqueidentifier,
SecurityProvider    uniqueidentifier,
{InDocLibrary}     bit,
{DocId}             uniqueidentifier,
{SiteFlags}        int,
Acl                 image,
AnonymousPermMask  bigint,
tp_ID               uniqueidentifier,
tp_Id               int,
tp_SiteAdmin        bit,
tp_IsActive         bit,
tp_Login            nvarchar(255),
tp_Email            nvarchar(255),
tp_Title            nvarchar(255),
tp_Notes            nvarchar(1023),
tp_ExternalTokenLastUpdated datetime,
tp_Token            image,
tp_Flags            int,
UserId              int,
{SiteSecurityVersion} bigint,
{PermCheckedAgainstUniqueList} int,
DraftOwnerId        int,
tp_Flags            bigint,
Level               tinyint,
{VirusVendorID}    int,
{VirusStatus}       int,
{VirusInfo}         nvarchar(255),

```

{ContentModifiedSince}	bit,
{ProgId}	nvarchar(255),
{DirName}	nvarchar(256),
{LeafName}	nvarchar(128),
{Type}	tinyint,
{DocLibRowId}	int,
Language	int,
{IsOldVersion}	bit;

Size: The size in bytes of the specified version of the document.

DocFlags: The document flags value of the document version.

TimeCreated: The date and time in Coordinated Universal Time (UTC) when the document for the specified version was last modified.

FullUrl: The store-relative form URL of the document.

{WebId}: The site identifier of the site (2) containing the document version.

FirstUniqueAncestorWebId: The site identifier whose security permissions are effective for the site (2) containing the specified document.

SecurityProvider: The identifier of the **COM** class of the **security provider (1)** for this site (2). This MUST be NULL for sites (2) using the native security implementation.

{InDocLibrary}: If the document is in a document library, this MUST be set to 1; otherwise, it MUST be set to zero.

{DocId}: The document identifier (2) of the requested document.

{SiteFlags}: The site collection flags value describing the configuration of the site collection containing the document.

Acl: The binary serialization of the ACL that is effective for the document.

AnonymousPermMask: Contains a 64-bit mask that specifies the permissions granted to an anonymous user.

tp_ID: The list identifier of the list (1) containing the document version. It MUST be NULL if this document is not in a list (1).

tp_Id: The user identifier for the specified user.

tp_SiteAdmin: Indicates whether the specified user is an administrator of the site collection. If yes, this MUST be set to 1; otherwise, this MUST be set to zero.

tp_IsActive: Indicates whether the specified user has created or modified any data in the site collection. If yes, this MUST be set to 1; otherwise, this MUST be set to zero.

tp_Login: The login name of the specified user.

tp_Email: The e-mail address of the specified user.

tp_Title: The display name of the specified user.

tp_Notes: Notes about the specified user.

tp_ExternalTokenLastUpdated: The date and time in Coordinated Universal Time (UTC) when the external group token value for the specified user was last updated. See [\[MS-WSSFO2\]](#), section [2.2.4.2](#).

tp_Token: An external group token value encoding information about external group membership for the user. This value can be NULL, indicating that this user has never visited any site (2) in the site collection. If this value is NULL, the value in tp_ExternalTokenLastUpdated MUST also be NULL. See [\[MS-WSSFO2\]](#), section [2.2.4.2](#).

tp_Flags: The user information flags value for the specified user.

UserId: The user identifier of the specified user. This can be NULL if the user has not been added as a member (2) to the site (2) whose permissions are in effect on the document.

{SiteSecurityVersion}: A version number incremented when changes are made to the site collection's permissions.

{PermCheckedAgainstUniqueList}: This MUST be zero.

DraftOwnerId: If the document version is a draft, the user identifier of the user who saved the first draft after the previous **published version** MUST be returned. If the document version is a published version and if there exists a draft whose version number is between that of the requested version and the next published version, the user identifier of the user who saved the first draft after the requested version MUST be returned. In all other cases, NULL MUST be returned.

tp_Flags: A **list flags** value describing the list (1) that contains the document.

Level: A publishing level value specifying the publish status of this document version.

{VirusVendorID}: The identifier of the virus scanner which processed this document version. This value MUST be NULL if the document has not been processed by a virus scanner.

{VirusStatus}: An enumerated type specifying the current virus check status of this document version. This value MUST be NULL if the requested document version has not been processed by a virus scanner. See [\[MS-WSSFO2\]](#), section 2.2.4.18 for a list of valid values.

{VirusInfo}: A string containing a provider-specific message returned by the virus scanner when it last processed the document version.

{ContentModifiedSince}: A bit indicating if the document version has been modified, depending on the value of **@ValidationType**.

It MUST be set to 1 if any of the following is true:

- The document is a **dynamic page**.
- **@ValidationType** is zero.
- **@ValidationType** is 1 and the value of **@ClientId** does not match the document identifier (2) in the store.
- **@ValidationType** is 2 and the time the specified document version was last modified does not match **@IfModifiedSince**.
- **@ValidationType** is 3 and the time the specified document version was last modified does not match **@IfModifiedSince** OR the value of **@ClientId** does not match the document identifier (2) in the store.

In all other cases, it MUST be set to zero.

{ProgId}: Specifies a preferred application designated to open the document.

{DirName}: The directory name of the requested document.

{LeafName}: The leaf name of the requested document.

{Type}: The **document store type** of this document version. See [MS-WSSFO2], section [2.2.2.4](#).

{DoclibRowId}: The document library row identifier for this document.

Language: The LCID of the language of the site (2), which is used to determine the display of the document on the front-end Web server. This parameter can be NULL.

{IsOldVersion}: A bit specifying whether this document is the current version or a historical version. It MUST be 1 for a historical version and zero for the current version. This value MAY be ignored by the protocol client.

3.1.4.44.5 Document Version Content Stream Result Set

The **Document Version Content Stream Result Set** contains the specified document version's document stream and associated metadata. It MUST return only if *@FetchType* is not set to "1", which means that it is not indicating an HTTP HEAD-only request. Also, it MUST NOT return if either the current version of the document is not found or the specified historical version of the document for the specified user is not found.

The **Document Version Content Stream Result Set** MUST return zero or one row.

The result set MUST return the document stream of the document if any of the following conditions is true:

- The value of **{ContentModifiedSince}** in the **Document Version Metadata Result Set** (section [3.1.4.44.4](#)) is "1".
- *@VirusVendorID* is NOT NULL and does not match the value of the identifier of the virus scanner that processed the document and the virus check status associated with the document is either unknown or set to zero.

Otherwise, the return code of "18" MUST be returned, and zero rows MUST be returned in this result set. The T-SQL syntax for the result set is as follows:

```
{Size}                int,  
{Content}             varbinary(max),  
{RbsResReference}    varbinary(800),  
{EtagVersion}        int,  
{Id}                 uniqueidentifier;  
{SetupPathVersion}   tinyint,  
{SetupPath}          nvarchar(255),  
{Dirty}              bit,  
DocFlags              int,  
{IsHistoryversion}   bit  
{Level}              tinyint;  
InternalVersion       int;
```

{Size}: The size of the document, in bytes.

{Content}: The document's content stream. For an uncustomized document or a document that has external storage, this MUST be NULL. Otherwise, if the content is larger than the value specified in the *@ChunkSize* parameter, only the first *@ChunkSize* bytes MUST be returned, and the front-end Web server can request individual chunks of content in a subsequent request.

{RbsResReference}: If remote BLOB storage is enabled and the document content is contained in a remote data store, this MUST be the remote BLOB storage identifier for the document content. If remote BLOB storage is disabled or the document content is not contained in a remote data store, this MUST be NULL. Further information about remote BLOB storage can be found in [\[MS-WSSO\]](#) section 2.1.2.3.8.

{ETagVersion}: An internal version counter incremented any time a change is made to this document, used for internal conflict detection. This value MUST be set to the current **ETagVersion** value for the document in the Docs View defined in [\[MS-WSSFO2\]](#) section 2.2.7.4.

{Id}: The document identifier (2) of the requested document.

{SetupPathVersion}: For an uncustomized document, this parameter governs the setup path location to which the setup path fragment is relative. This value MUST be NULL if the document does not exist, and it is undefined for a document that was never uncustomized. The following are all valid values.

Value	Description
"2"	Relative to the install location of Windows® SharePoint® Services 2.0 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\60)
"3"	Relative to the install location of Windows® SharePoint® Services 3.0 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\12).
"4"	Relative to the install location of Microsoft® SharePoint® Foundation 2010 on the front-end Web server (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\14).

{SetupPath}: For a document that has never been uncustomized, this contains the setup path fragment relative to the base setup path where the content stream of this document can be found, as described in **{SetupPathVersion}**. Otherwise, this parameter MUST be NULL.

The second part of it is the setup user. For the author of the document.

{Dirty}: MUST be "1" if this document has had "content stream needs dependency update" processing performed; otherwise, this MUST be zero. If the document does not have a content stream, the value is implementation-dependent and MUST be ignored.

DocFlags: A Doc Flags value as defined in [\[MS-WSSFO2\]](#) section 2.2.2.3, describing the document.

{IsHistoricalVersion}: A bit specifying whether this document is the current version or a historical version. It MUST be "1" for a historical version and zero ("0") for the current version.

{Level}: The publishing level value of this document.

InternalVersion: An **integer** value specifying the implementation-related internal version number of this version of the document.

3.1.4.44.6 Site Metadata Result Set

The **Site Metadata Result Set** contains metadata for the site (2) containing the specified document. The **Site Metadata Result Set** MUST return only if the input parameter **@PageView** is not NULL, as part of a series of result sets describing list view webpage document metadata.

The **Site Metadata Result Set** is defined in [\[MS-WSSFO2\]](#) section 2.2.5.22.

3.1.4.44.7 Event Receivers Result Set

The **Event Receivers Result Set** contains information about the **event receivers** defined for the site collection containing the specified document.

The **Event Receivers Result Set** is part of a series of result sets describing list view webpage-related metadata. It MUST return ONLY if the input parameter **@PageView** is not NULL, and it will not return if **proc_FetchDocForHttpGet** returns a return code of 1271, indicating that the site collection was locked.

The **Event Receivers Result Set** MUST contain one row per event receiver registered for the site collection. The **Event Receivers Result Set** is defined in [\[MS-WSSFO2\]](#) section 2.2.5.9.

3.1.4.44.8 Web Event Receivers Result Set

The **Web Event Receivers Result Set** contains information about the event receivers defined for the site (2) containing the specified document.

The **Web Event Receivers Result Set** is part of a series of result sets describing list view webpage-related metadata. It MUST return ONLY if the input parameter **@PageView** is not NULL, and it will NOT return if **proc_FetchDocForHttpGet** returns a return code of 1271, indicating that the site collection was locked.

The **Web Event Receivers Result Set** MUST contain one row per event receiver registered for the site (2). The **Web Event Receiver Result Set** is defined in [\[MS-WSSFO2\]](#) section 2.2.5.9.

3.1.4.44.9 Site Features List Result Set

The **Site Features List Result Set** returns information about available features. If the **Site Features List Result Set** returns, it MUST return twice: first for site collection features, and then for site features, for the site and site collection that contain the specified document.

The **Site Features List Result Set** MUST return only if **@PageView** is not NULL, and it will NOT return if **proc_FetchDocForHttpGet** returns a code of 1271, indicating that the site collection was locked.

The **Site Features List Result Set** is defined in [\[MS-WSSFO2\]](#) section 2.2.5.21.

3.1.4.44.10 Web Parts Metadata, Nonpersonalized Result Set

The **Web Parts Metadata, Nonpersonalized Result Set** contains the core metadata about the Web parts appearing on the specified document.

The **Web Parts Metadata, Nonpersonalized Result Set** is part of a series of result sets describing list view webpage-related metadata. It MUST return ONLY if the input **@PageView** is not NULL, and it will NOT return if **proc_FetchDocForHttpGet** returns a code of 1271, indicating that the site collection was locked.

The **Web Parts Metadata, Nonpersonalized Result Set** MUST contain one row per Web Part. The result set is specified in [\[MS-WSSFO2\]](#) section 3.1.5.19.18.

3.1.4.44.11 List Metadata Result Set

The **List Metadata Result Set** contains the metadata for the lists (1) associated with the Web Parts that are included on the specified document.

The **List Metadata Result Set** returns ONLY if there are such Web Parts (at least one row returns in the previously returned result set).

The **List Metadata Result Set** is part of a series of result sets describing list view webpage-related metadata. It MUST be returned only if the input parameter **@PageView** is not NULL, and it MUST NOT be returned if **proc_FetchDocForHttpGet** returns a code of 1271, indicating that the site collection was locked.

The **List Metadata Result Set** MUST return one row for each associated list (1). The **List Metadata Result Set** is specified in [\[MS-WSSFO2\]](#), section [2.2.5.12](#).

3.1.4.44.12 List Event Receivers Result Set

The **List Event Receivers Result Set** contains the event receivers registered on the lists associated with the Web Parts that appear on the specified document.

The **List Event Receivers Result Set** MUST return only if there are such lists (the result set in section [3.1.4.44.11](#) returns with at least one row).

The **List Event Receivers Result Set** MUST contain one row per event receiver registered. **The List Event Receivers Result Set** can be empty. The **List Event Receivers Result Set** is specified in [\[MS-WSSFO2\]](#) section 3.1.5.20.

3.1.4.44.13 List Security Information Result Set

The **List Security Information Result Set** contains permissions information for the lists (1) associated with the Web Parts that appear on the specified document.

The **List Security Information Result Set** MUST return only if such lists exist (the result set in section [3.1.4.44.11](#) returns with at least one row).

The **List Security Information Result Set** returns one row per each unique scope associated with the lists. It is specified in [\[MS-WSSFO2\]](#) section 3.1.5.19.21.

3.1.4.44.14 Site Collection Custom Actions Result Set

The **Site Collection Custom Actions Result Set** MUST return 1 row for each **custom action** defined for the site collection containing the specified document.

The **Site Collection Custom Actions Result Set** is part of a series of result sets describing list view webpage-related metadata. It MUST return ONLY if the input parameter **@PageView** is not NULL, and it will NOT return if **proc_FetchDocForHttpGet** returns a return code of 1271, indicating that the site collection was locked.

If there are no custom actions defined for the site collection, this result set MUST NOT return any rows. This result set is specified in [\[MS-WSSFO2\]](#), section [2.2.5.2](#).

3.1.4.44.15 Site Custom Actions Result Set

The **Site Custom Actions Result Set** MUST return 1 row for each custom action defined for the site (2) containing the specified document.

The **Site Custom Actions Result Set** is part of a series of result sets describing list view webpage-related metadata. It MUST return ONLY if the input parameter **@PageView** is not NULL, and it will NOT return if **proc_FetchDocForHttpGet** returns a return code of 1271, indicating that the site collection was locked.

If there are no custom actions defined for the site (2), this result set MUST NOT return any rows. This result set is specified in [\[MS-WSSFO2\]](#), section [2.2.5.2](#).

3.1.4.44.16 List Custom Actions Result Set

The **List Custom Actions Result Set** MUST return 1 row for each custom action defined for the lists (1) associated with the Web Parts that appear on the specified document.

The **List Custom Actions Result Set** MUST return only if there are such lists (the result set in section [3.1.4.44.11](#) returns with at least one row).

If there are no custom actions defined for the lists (1) returned by the result set in section [3.1.4.44.11](#), this result set MUST NOT return any rows. This result set is specified in [\[MS-WSSFO2\]](#), section [2.2.5.2](#).

3.1.4.44.17 List Web Parts Result Set

The **List Web Parts Result Set** contains information about the Web parts related to the lists (1) associated with the specified document.

The **List Web Parts Result Set** MUST return only if such lists (1) exist (the result set in section [3.1.4.44.11](#) returns with at least one row).

The **List Web Parts Result Set** MUST contain one row per Web Part registered for each list (1). The **List Web Parts Result Set** can be empty. The **List Web Parts Result Set** is specified in [\[MS-WSSFO2\]](#), section [2.2.5.13](#).

3.1.4.44.18 NULL Content Type Order Result Set

The **NULL Content Type Order Result Set** is a special case of the **Content Type Order Result Set**, as defined in [\[MS-WSSFO2\]](#) section 3.1.5.19.26, where the columns have fixed placeholder values.

The **NULL Content Type Order Result Set** MUST return only if **@PageView** is not NULL, and it will NOT return if **proc_FetchDocForHttpGet** returns a code of 1271, indicating that the site collection was locked.

If the **NULL Content Type Order Result Set** returns, it must return one row. The T-SQL syntax for the result set is as follows:

```
{CurrentFolderURL}          varchar,  
{MetaInfo}                  varbinary(max)
```

{CurrentFolderURL}: This MUST be empty string.

{MetaInfo}: This MUST be NULL.

3.1.4.44.19 Navigation Context Security Information Result Set

The **Navigation Context Security Information Result Set** contains security information about the site (2) containing the specified document and about all sites (2) in its navigation hierarchy.

The **Navigation Context Security Information Result Set** MUST return only upon successful execution, if @PageView is not NULL and the information is not larger than 1,800 bytes.

If the **Navigation Context Security Information Result Set** returns, it MUST return one row for each unique scope in the navigation hierarchy of the site (2), excluding the scope for the site (2) itself.

The **Navigation Context Security Information Result Set** is specified in [\[MS-WSSFO2\]](#) section 3.1.5.19.28.

3.1.4.44.20 NULL Navigation Context Security Information Result Set

The **NULL Navigation Context Security Information Result Set**, with NULL values in three unnamed columns, returns to indicate that the navigation context security information is larger than 1,800 bytes.

The **NULL Navigation Context Security Information Result Set** MUST be returned only upon successful execution if @PageView is not NULL and the security information about the site (2) or the parent site in the navigation hierarchy for the site (2) was larger than 1,800 bytes.

If the **NULL Navigation Context Security Information Result Set** returns, it MUST return one row, as specified in [\[MS-WSSFO2\]](#) section 3.1.5.19.29.

3.1.4.44.21 Empty Navigation Context Security Information Result Set

The **Empty Navigation Context Security Information Result Set**, holding zero rows with a single, unnamed NULL column, is returned to indicate that the navigation context security information is not available or is not up-to-date. The **Empty Navigation Context Security Information Result Set** MUST return only upon successful execution, and if neither the [Navigation Context Security Information Result Set](#) (section [3.1.4.44.19](#)) or the [NULL Navigation Context Security Information Result Set](#) (section [3.1.4.44.20](#)) is returned.

If the **Empty Navigation Context Security Information Result Set** returns, it MUST return zero rows, as specified in [\[MS-WSSFO2\]](#) section 3.1.5.19.30.

3.1.4.45 proc_FileFragmentPermissionCheck

The **proc_FileFragmentPermissionCheck** stored procedure is called to ensure a document exists and has an appropriate lock for use when updating file fragments associated with it. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_FileFragmentPermissionCheck (  
    @SiteId                uniqueidentifier,  
    @DocId                 uniqueidentifier,  
    @UserId                int,  
    @DocVersion            int,  
    @ContentVersion        int,  
    @Partition             tinyint,  
    @MaxSizeInBytes        bigint
```

);

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

@DocId: The document identifier (2) of the document to update.

@UserId: The user identifier of the user requesting the operation.

@DocVersion: The internal version number of the document to update.

@ContentVersion: The content version of the document to update.

@Partition: The Identifier for the file fragment partition to update.

@MaxSizeInBytes: A maximum size of allowed data, in bytes.

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

Value	Description
0	Successful execution.
2	The current document with document identifier (2) equal to @DocId and site collection identifier equal to @SiteId was not found.
5	There is a short-term lock on the file and either: the owner of the lock is NULL or both the DocFlags does not contain flag (131072) DF_SHARED_LOCK and the owner of the lock is not equal to @UserId .
32	The current internal version of the document is NULL or is not equal to @DocVersion , or the current internal content version of the document is NULL or @ContentVersion is null or the current internal content version is not equal to @ContentVersion .
164	The amount of data, in bytes, in the file fragment partition specified by @Partition exceeds @MaxSizeInBytes .

Result sets: MUST NOT return any result sets.

3.1.4.46 `proc_FindDocs`

The **proc_FindDocs** stored procedure is called to determine if one or more documents exist in a site collection. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_FindDocs(  
    @SiteId          uniqueidentifier,  
    @DirName1       nvarchar(256) = NULL,  
    @LeafName1      nvarchar(256) = NULL,  
    @DirName2       nvarchar(256) = NULL,  
    @LeafName2      nvarchar(256) = NULL,  
    @DirName3       nvarchar(256) = NULL,  
    @LeafName3      nvarchar(256) = NULL,  
    @DirName4       nvarchar(256) = NULL,  
    @LeafName4      nvarchar(256) = NULL,  
    @DirName5       nvarchar(256) = NULL,  
    @LeafName5      nvarchar(256) = NULL,  
    @DirName6       nvarchar(256) = NULL,  
    @LeafName6      nvarchar(256) = NULL,
```

```

        @DirName7          nvarchar(256) = NULL,
        @LeafName7        nvarchar(256) = NULL,
        @DirName8          nvarchar(256) = NULL,
        @LeafName8        nvarchar(256) = NULL,
        @RequestGuid      uniqueidentifier = NULL OUTPUT
    );

```

@SiteId: The site collection identifier of the site collection which contains the specified documents to be searched.

@DirName1: The store-relative form folder URL that contains the document specified by @LeafName1.

@LeafName1: The document name in the folder specified by **@DirName1** that is to be retrieved.

@DirName2: The store-relative form folder URL that contains the document specified by @LeafName2.

@LeafName2: The document name in the folder specified by **@DirName2** that is to be retrieved.

@DirName3: The store-relative form folder URL that contains the document specified by @LeafName3.

@LeafName3: The document name in the folder specified by **@DirName3** that is to be retrieved.

@DirName4: The store-relative form folder URL that contains the document specified by @LeafName4.

@LeafName4: The document name in the folder specified by **@DirName4** that is to be retrieved.

@DirName5: The store-relative form folder URL that contains the document specified by @LeafName5.

@LeafName5: The document name in the folder specified by **@DirName5** that is to be retrieved.

@DirName6: The store-relative form folder URL that contains the document specified by @LeafName6.

@LeafName6: The document name in the folder specified by **@DirName6** that is to be retrieved.

@DirName7: The store-relative form folder URL that contains the document specified by @LeafName7.

@LeafName7: The document name in the folder specified by **@DirName7** that is to be retrieved.

@DirName8: The store-relative form folder URL that contains the document specified by @LeafName8.

@LeafName8: The document name in the folder specified by **@DirName8** that is to be retrieved.

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

Return values: An integer that MUST be zero.

Result sets: MUST return 1 result set as described in the following section.

3.1.4.46.1 Found Docs Result Set

The **Found Docs Result Set** returns the full URL of each document specified in the request that was found in the site collection. The **Found Docs Result Set** returns one row for each found document. The T-SQL syntax for the result set is as follows:

```
FullName          nvarchar(384);
```

FullName: The store-relative URL for the specified document.

3.1.4.47 proc_FinishUndirtyList

The **proc_FinishUndirtyList** stored procedure is called to reset the dirty status on a list (1) upon completion of a link fixup operation. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_FinishUndirtyList (
    @WebId          uniqueidentifier,
    @ListId         uniqueidentifier,
    @CacheParseId  uniqueidentifier,
    @ListFlag       int,
    @RequestGuid    uniqueidentifier = NULL OUTPUT
);
```

@WebId: The site identifier of the site (2) which contains the list (1). MUST NOT be NULL.

@ListId: The list identifier of the list (1) to reset the dirty status. MUST NOT be NULL.

@CacheParseId: Used for concurrency detection when two different requests attempt to reset the dirty status on a list (1) or its documents at the same time. This MUST be the parameter passed in the call to **proc_StartUndirtyList** (section [3.1.4.92](#)) at the start of this link fixup operation. If a concurrent link fixup operation started after this link fixup operation started, the stored procedure will have no effect. The latest link fixup operation will have to reset the dirty status. MUST NOT be NULL.

@ListFlag: An integer whose value MUST be 1.

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

Return values: An integer that the protocol client MUST ignore.

Result sets: MUST NOT return any result sets.

3.1.4.48 proc_GenerateUniqueFileName

The **proc_GenerateUniqueFileName** stored procedure is called to generate a unique file name from a given base name and extension. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GenerateUniqueFileName (
    @SiteId          uniqueidentifier,
    @BaseUrl         nvarchar(260),
    @Extension       nvarchar(10),
    @MaxAttempts     int,
    @RequestGuid     uniqueidentifier = NULL OUTPUT
);
```

@SiteId: The site collection identifier of the site collection which is to contain the specified file.

@BaseUrl: The desired base name in store-relative form of the file.

@Extension: The desired extension of the file.

@MaxAttempts: The maximum number of attempts to generate a unique file name.

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

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

Value	Description
0	Successful execution.
3	No unique file name can be generated within the specified @MaxAttempts.

Result sets: MUST return 1 result set if the return code is zero and MUST NOT return any result sets if the return value is not zero.

3.1.4.48.1 Unique File Name Result Set

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

```
{File Name}          nvarchar(385)
```

{File Name}: The URL in store-relative form, generated from the given base file name and extension, which is unique in the site collection.

3.1.4.49 proc_GetAllAttachmentsInfo

The **proc_GetAllAttachmentsInfo** stored procedure is invoked to retrieve information about attachments to one or all items in a specified list (1).

```
PROCEDURE proc_GetAllAttachmentsInfo(  
    @SiteID                uniqueidentifier,  
    @WebID                 uniqueidentifier,  
    @ListID                uniqueidentifier,  
    @ItemID                int,  
    @RequestGuid           uniqueidentifier = NULL    OUTPUT  
);
```

@SiteID: The site collection identifier of the site collection containing the List.

@WebID: The site identifier of the site containing the list.

@ListID: The list identifier of the List containing the attachments.

@ItemID: If set to -1, indicates information is requested for attachments to all items in the list. Otherwise, information is requested only for attachments to the specific item identified by **@ItemID**.

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

Return values: The **proc_GetAllAttachmentsInfo** stored procedure returns an integer return code, which MUST be zero.

Result sets: The **proc_GetAllAttachmentsInfo** stored procedure MUST return one of two result sets.

3.1.4.49.1 List Attachments Result Set

The **List Attachments Result Set** returns a list of the attachments (attached documents) associated with any item in the specified list (1). The **List Attachments Result Set** MUST be returned when @ItemID is set to -1, and MUST return one row for each attachment in the list (1).

DocItemId	int,
LeafName	nvarchar(128),
Id	uniqueidentifier,
ETagVersion	int,
Acl	image,
AnonymousPermMask	bigint;

DocItemId: Contains an identifier for the parent document (list item) within the list (1).

LeafName: Contains the leaf name of the attachment document.

Id: Contains the document identifier (2) of the attachment document.

ETagVersion: Contains the internal version number of the attachment document.

Acl: Contains the binary serialization of the WSS ACL Format ACL for the attachment document as specified in [\[MS-WSSFO2\]](#) section 2.2.4.6. This is either explicitly defined, or inherited from the parent object.

AnonymousPermMask: Contains a WSS Rights Mask indicating the **rights** granted to an anonymous user, or to a user who has no specific rights to the document, as specified in [\[MS-WSSFO2\]](#) section 2.2.2.14.

3.1.4.49.2 Document Attachments Result Set

The **Document Attachments Result Set** returns a list of attachments for the document specified by @ItemID. The **Document Attachments Result Set** MUST be returned when @ItemID is NOT set to -1. The **Document Attachments Result Set** MUST return one row for each attachment to the specified document.

Id	uniqueidentifier,
LeafName	nvarchar(128);

Id: Contains the document identifier for the attachment document.

LeafName: Contains the leaf name of the attachment document.

3.1.4.50 proc_GetAllListsPlusProperties

The **proc_GetAllListsPlusProperties** stored procedure is called to get additional list (1) properties which are stored in the ListsPlus table. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_GetAllListsPlusProperties (
    @ListId          uniqueidentifier
);

```

@ListId: The list identifier of the list (1) whose properties need to be fetched.

Return values: An integer that the protocol client MUST ignore.

Result sets: MUST return the **AllListsPlus Properties Result Set** (section [3.1.4.50.1](#)).

3.1.4.50.1 AllListsPlus Properties Result Set

The **AllListsPlus Properties Result Set** returns no rows if the list (1) specified does not exist, else it returns one row. The T-SQL syntax for the result set is as follows:

```

ValidationFormula    nvarchar(1024),
ValidationMessage    nvarchar(1024);

```

ValidationFormula: A Unicode string that can be used to perform custom validation rules prior to the list (1) being updated. This MUST be an expression as specified by [\[MS-WSSTS\]](#) section 2.9.

ValidationMessage: A Unicode string (or strings) that are suitable to display in a user interface when the list (1) fails validation based on ValidationFormula.

3.1.4.51 proc_GetChanges

The **proc_GetChanges** stored procedure is called to get a range of events from the change log specified by the parameters. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_GetChanges (
    @SiteId                uniqueidentifier,
    @WebId                  uniqueidentifier,
    @ListId                 uniqueidentifier,
    @ChangeTime             datetime,
    @ChangeNumber           bigint,
    @ChangeTimeEnd          datetime,
    @ChangeNumberEnd        bigint,
    @ObjectTypeMask         int,
    @EventTypeMask          int,
    @MaxChanges             bigint,
    @IgnoreChangeNumberIfNotFound bit,
    @RequestGuid            uniqueidentifier = NULL OUTPUT
);

```

@SiteId: The site collection identifier of the site collection with which the events are associated. If this parameter is NULL, events from all site collections MUST be included.

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

@ListId: The list identifier of the list (1) with which the events are associated. If this parameter is NULL, then all events in the change log that have an empty site identifier MUST be included.

@ChangeTime: A time stamp in Coordinated Universal Time (UTC). This parameter defines the lower bound time stamp of the events returned from the change log. If **@ChangeTime** is NULL and

@ChangeNumber is NULL then the first event (1) in the change log MUST begin the range of events included in the result. If **@ChangeTime** is NULL and **@ChangeNumber** is NOT NULL, then the event (1) with change number equal to **@ChangeNumber**, or the first event (1) in the change log if no such event (1) exists, MUST begin the range of events (1) included. If **@ChangeTime** is not NULL and **@ChangeNumber** is NULL, then the first event (1) that occurred before **@ChangeTime**, or the first event (1) in the change log if no such event (1) exists, MUST begin the range of events (1) included in the result. If **@ChangeTime** is NOT NULL and **@ChangeNumber** is NOT NULL, then the event (1) with the **change log identifier** that matches **@ChangeNumber** MUST begin the range of events (1) included in the result unless the time stamp of the event (1) does not match **@ChangeTime**, in which case the beginning event (1) in the result is determined using the **@IgnoreChangeNumberIfNotFound** parameter.

@ChangeNumber: The lower bound change log identifier of the events to be included in the result.

@ChangeTimeEnd: A time stamp in Coordinated Universal Time (UTC). This parameter defines the upper bound time stamp of the events returned from the change log. If this parameter is NULL, then the upper bound will be the time stamp of the most recent event (1) in the change log.

@ChangeNumberEnd: The upper bound change log identifier of the events (1) returned from the change log. If this parameter is NOT NULL, then the **@ChangeTimeEnd** parameter MUST be ignored.

@ObjectTypeMask: A bit mask that specifies the type of objects upon which an event (1) had happened. Valid values for this flag are defined in **Event Object Type Flags** (section [2.2.3.1](#)).

@EventMask: A bit mask that specifies the type of an event (1). Valid values for this flag are defined in **Event Type Flags** (section [2.2.3.2](#)).

@MaxChanges: The maximum number of events (1) to return.

@IgnoreChangeNumberIfNotFound: If **@IgnoreChangeNumberIfNotFound** is zero, and **@ChangeNumber** and **@ChangeTime** are both not NULL, and they do not correspond to the same event (1) in the change log, then **@ChangeNumber** MUST be ignored in the determination of the set of events (1) in the change log. If **@IgnoreChangeNumberIfNotFound** is one and **@ChangeNumber** and **@ChangeTime** are both not NULL, then **@ChangeNumber** MUST NOT be ignored in the determination of the set of events (1) in the change log. In that case, if **@ChangeNumber** and **@ChangeTime** do not correspond to the same event (1), then zero rows must be returned in the **EventInformation Result Set** (section [3.1.4.51.1](#)).

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

Return values: An integer that the protocol client MUST ignore.

Result sets: MUST return two result sets in the order of the following sections.

3.1.4.51.1 EventInformation Result Set

The **EventInformation Result Set** returns the event (1) that has the smallest change log identifier in the change log. The **EventInformation Result Set** MUST return one row in the result set if an event (1) exists in the change log or zero rows if no event (1) exists in the change log. The T-SQL syntax for the result set is as follows:

```
EventTime    datetime,  
Id           bigint;
```

EventTime: A time stamp in Coordinated Universal Time (UTC) that specifies the time when this event (1) occurred.

Id: The change log identifier of this event (1).

3.1.4.51.2 EventDetails Result Set

The **EventDetails Result Set** returns details of events that satisfy the input parameters. The **EventDetails Result Set** MUST NOT contain more than **@MaxChanges** rows. The T-SQL syntax for the result set is as follows:

```
EventTime          datetime,
Id                 bigint,
SiteId            uniqueidentifier,
WebId             uniqueidentifier,
ListId            uniqueidentifier,
ItemId            int,
DocId             uniqueidentifier,
Guid0             uniqueidentifier,
Int0              int,
ContentTypeId     varbinary(512),
ItemFullUrl       nvarchar(266),
EventType         int,
ObjectType        int,
TimeLastModified  datetime,
Int1              int;
```

EventTime: A time stamp in Coordinated Universal Time (UTC) that specifies when this event (1) occurred.

Id: The change log identifier of this event (1).

SiteId: This value is a change log **SiteId** (section [2.2.2.11](#)).

WebId: This value is a change log **WebId** (section [2.2.2.12](#)).

ListId: This value is a change log **ListId** (section [2.2.2.1](#)).

ItemId: This value is a change log **ItemId** (section [2.2.2.2](#)).

DocId: This value is a change log **DocId** (section [2.2.2.3](#)).

Guid0: This value is a change log **Guid0** (section [2.2.2.4](#)).

Int0: This value is a change log **Int0** (section [2.2.2.5](#)).

ContentTypeId: This value is a change log **ContentTypeId** (section [2.2.2.6](#)).

ItemFullUrl: This value is a change log **ItemFullUrl** (section [2.2.2.7](#)).

EventType: A bit mask that specifies the type of an event (1). Valid values of this flag are defined in **Bit Fields and Flag Structures** (section [2.2.3](#)).

ObjectType: A bit mask that specifies the type of object upon which an event (1) had happened. Valid values of this flag are defined in **Simple Data Types and Enumerations** (section [2.2.1](#)).

TimeLastModified: This value is specified in **Change Log TimeLastModified** (section [2.2.2.8](#)).

Int1: This value is a change log **Int1** (section [2.2.2.10](#)).

3.1.4.52 `proc_GetChangeToken`

The **proc_GetChangeToken** stored procedure is called to return either the time stamp or the change log identifier of the specified event (1) from the change log. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetChangeToken (
    @ChangeTime    datetime,
    @ChangeNumber  bigint,
);
```

@ChangeTime: A time stamp in Coordinated Universal Time (UTC) that specifies the date and time of the change.

@ChangeNumber: An integer specifying the number of the change. MUST be NULL if **@ChangeTime** is not NULL. MUST NOT be NULL if **@ChangeTime** is NULL.

Return values: An integer that MUST be zero.

Result sets: If **@ChangeTime** is not NULL, this stored procedure MUST return the **Change Token Result Set** (section [3.1.4.52.1](#)). If **@ChangeNumber** is not NULL, this stored procedure MUST return the **Change Token Time Result Set** (section [3.1.4.52.2](#)).

3.1.4.52.1 Change Token Result Set

The **Change Token Result Set** returns the change log identifier of the latest event (1) from the change log corresponding to the datetime specified by **@ChangeTime**. The **Change Token Identifier Result Set** MUST return one row if an event (1) is found or zero rows if no event (1) is found. The T-SQL syntax for the result set is as follows:

```
Id          bigint NOT NULL;
```

Id: The change log identifier of this event (1).

3.1.4.52.2 Change Token Time Result Set

The **Change Token Time Result Set** returns the time stamp in Coordinated Universal Time (UTC) of the event (1) from the change log corresponding to the value specified by **@ChangeNumber**. The **Change Token Time Result Set** MUST return one row if an event (1) is found or zero rows if no event (1) is found. The T-SQL syntax for the result set is as follows:

```
EventTime   datetime;
```

EventTime: A time stamp in Coordinated Universal Time (UTC) that specifies when this event (1) occurred.

3.1.4.53 `proc_GetCurrent`

The **proc_GetCurrent** stored procedure is called to return the time stamp and change log identifier of the latest event (1) from the change log. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_GetCurrent(
    @RequestGuid          uniqueidentifier = NULL OUTPUT
);

```

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

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST return one result set as specified in the following sections.

3.1.4.53.1 EventInformation Result Set

The **EventInformation Result Set** returns the time stamp and change log identifier of the most current event (1) from the change log. The **EventInformation Result Set** MUST return one row if an event (1) is found or zero rows if no event (1) is found. The T-SQL syntax for the result set is as follows:

```

EventTime      datetime,
Id              bigint;

```

EventTime: A time stamp in Coordinated Universal Time (UTC) that specifies when this event (1) occurred.

Id: The change log identifier of this event (1).

3.1.4.54 proc_GetDocIdUrl

The **proc_GetDocIdUrl** stored procedure is called to retrieve the identifier of a specified document. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_GetDocIdUrl(
    @SiteID          uniqueidentifier,
    @DocDirName      nvarchar(256),
    @DocLeafName     nvarchar(128),
    @DocID           uniqueidentifier OUTPUT,
    @RequestGuid     uniqueidentifier = NULL OUTPUT
);

```

@SiteID: The site collection identifier of the site collection which contains the specified document.

@DocDirName: The directory name of the specified document.

@DocLeafName: The leaf name of the requested Document.

@DocID: An output parameter containing the identifier of the specified Document if execution is successful. If the specified Document does not exist, the **proc_GetDocIdUrl** MUST NOT set the value of @DocID.

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

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

Value	Description
0	Successful execution.
2	Cannot find the specified document in the site collection, or the site collection does not exist.

Result sets: MUST NOT return any result sets.

3.1.4.55 **proc_GetFileFragmentsById**

The **proc_GetFileFragmentsById** stored procedure is called to fetch file fragments for a document with a specified File Fragment Identifier value. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetFileFragmentsById (
    @SiteId                uniqueidentifier,
    @DocId                 uniqueidentifier,
    @Partition             tinyint,
    @Id                    bigint,
    @TopRows               bigint,
    @GetAfter              bit,
    @GetOnly               bit,
    @DocVersion            int,
    @ContentVersion        int
);
```

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

@DocId: The document identifier (2) of the document associated with the file fragments being fetched.

@Partition: The identifier for a file fragment partition of the file fragment partition to which the file fragment being fetched belong.

@Id: The File Fragment Identifier value to verify against when fetching data.

@TopRows: Specifies the number of file fragments to fetch.

@GetAfter: Specifies whether file fragments with File Fragment Identifier greater than **@Id** **MUST** be fetched.

@GetOnly: Specifies whether just the file fragment with File Fragment Identifier equal to **@Id** **MUST** be fetched.

@DocVersion: The internal version number of the document to read from. If the current internal version number of the document is NULL or not equal to **@DocVersion**, the **File Fragment Result Set** (section [2.2.6.7](#)) **MUST** contain zero rows.

@ContentVersion: The content version of the document to read from. If **@ContentVersion** is NULL or the current content version number of the document is not NULL or not equal to **@ContentVersion**, the **File Fragment Result Set** **MUST** contain zero rows.

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

Value	Description
0	Successful execution.
2	The document specified by @DocId was not found.
32	The current internal version number of the document is NULL or is not equal to @DocVersion , or the current content version of the document is null or @ContentVersion is null or the current content version is not equal to @ContentVersion .
87	@GetAfter is 1 and @GetOnly is 1.
1398	@Id is greater than or equal to the last committed file fragment.

Result sets: MUST return one **File Fragment Result Set**.

3.1.4.56 **proc_GetFileFragmentsByTag**

The **proc_GetFileFragmentsByTag** stored procedure is called to fetch file fragments for a document with a specified file fragment tag and file fragment identifier value. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_GetFileFragmentsByTag (
    @SiteId          uniqueidentifier,
    @DocId           uniqueidentifier,
    @Partition       tinyint,
    @Tag             varbinary(40),
    @TopRows         bigint,
    @AfterId         bigint,
    @GetLatest       bit,
    @GetAfter        bit,
    @DocVersion      int,
    @ContentVersion  int
);

```

@SiteId: The site collection identifier of the site collection which contains the file fragments being fetched.

@DocId: The document identifier (2) of the document associated with the file fragments being fetched.

@Partition: The Identifier for a file fragment partition of the file fragment partition to which the file fragment being fetched belongs.

@Tag: The file fragment tag value to verify against when fetching data.

@TopRows: Specifies the amount of rows to fetch in conjunction with the **@Tag** parameter. If **@GetLatest** is NULL or zero and **@GetAfter** is NULL or zero, only **@TopRows** number of file fragments MUST be fetched.

@AfterId: The file fragment identifier value to verify against when fetching data.

@GetLatest: If **@GetLatest** is 1, the file fragment with maximum value of file fragment identifier and file fragment tag equal to **@Tag** MUST be fetched.

@GetAfter: If **@GetAfter** is 1, **@TopRows** number of file fragments with file fragment identifier greater than **@AfterId** and file fragment tag equal to **@Tag** MUST be fetched.

@DocVersion: The internal version number of the document to read from. If the current internal version number of the document is NULL or not equal to **@DocVersion**, the **File Fragment Result Set** (section [2.2.6.7](#)) MUST contain zero rows.

@ContentVersion: The content version of the document to read from. If **@ContentVersion** is NULL or the current content version number of the document is not NULL or not equal to **@ContentVersion**, the **File Fragment Result Set** MUST contain zero rows.

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

Value	Description
0	Successful execution.
2	The document specified by @DocId was not found.
32	The current internal version number of the document is NULL or is not equal to @DocVersion , or the current content version of the document is null or @ContentVersion is null or the current content version is not equal to @ContentVersion .
87	@GetAfter is 1 and @GetLatest is 1.
1398	@AfterId is greater than or equal to the last committed file fragment.

Result sets: MUST return 1 **File Fragment Result Set**.

3.1.4.57 **proc_GetFullLinkInfoForSingleDoc**

The **proc_GetFullLinkInfoForSingleDoc** stored procedure is called to return information of all the links for a single document. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetFullLinkInfoForSingleDoc (
    @DocSiteId                uniqueidentifier,
    @DocDirName                nvarchar(256),
    @DocLeafName               nvarchar(128),
    @UserId                    int,
    @AttachmentsFlag           tinyint,
    @MaxCheckinLevel           tinyint,
    @GetWebListForNormalization bit,
    @RequestGuid               uniqueidentifier = NULL OUTPUT
);
```

@DocSiteId: The site collection identifier of the site collection containing the document. MUST NOT be NULL.

@DocDirName: The directory name of the directory containing the document. MUST NOT be NULL.

@DocLeafName: The leaf name of the document. MUST NOT be NULL.

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

@AttachmentsFlag: Bit that governs the type of security checks which SHOULD be performed by a stored procedure on this document's URL, based on whether it appears to be an attachment. A value which MUST be listed in the following table.

Value	Description
0	The URL does not appear to be an attachment.
1	The URL is an attachment file. The directory name of the document has the string "Attachments" as its next-to-last path segment, and a 32-bit base-10 signed integer as the last path segment that is referring to the item identifier to which this file is attached and where the permissions will be checked. For example, Announcements/Attachments/17/file1.txt.
2	The URL is a folder containing list item attachments. For example, Announcements/Attachments/17.
3	The URL is the folder containing the list attachments. The last path segment of the URL is the string "Attachments". For example, "Announcements/Attachments".

@MaxCheckinLevel: A Publishing Level Type, (as specified in [\[MS-WSSFO2\]](#) section 2.2.2.6) value specifying the maximum publishing level of the links to return.

@GetWebListForNormalization: Bit flag indicating that the **Web List For Normalization Result Set** (section [3.1.4.57.1](#)) SHOULD be returned.

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

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

Value	Description
0	Success
2	The Document store type of the specified object is not zero and it is not an item in a list (1) or document library.

Result sets: The stored procedure might return the **Web List For Normalization Result Set**, and MUST return the **Individual URL Security Result Set** (section [3.1.4.57.3](#)), the **Document Link Information Result Set** (section [3.1.4.57.4](#)), and the **Document Setup Path Result Set** (section [3.1.4.57.5](#)).

3.1.4.57.1 Web List For Normalization Result Set

If the **@GetWebListForNormalization** flag is set to **true**, the **Web List For Normalization Result Set** returns a list of **Fully qualified URL** for subsites of the specified site (2). The result set will contain zero or more rows. The T-SQL syntax for the result set is as follows:

```
FullUrl          nvarchar(256);
```

FullUrl: Full URL of a subsite.

3.1.4.57.2 NULL Individual URL Security Result Set

The **NULL Individual URL Security Result Set** MUST only be returned if the specified document location is NOT contained within a list (1) or document library. It MUST contain a single row. The **NULL Individual URL Security Result Set** is defined in [\[MS-WSSFO2\]](#) section 2.2.5.14.

3.1.4.57.3 Individual URL Security Result Set

The **Individual URL Security Result Set** contains security information about the specified document. If the document does not exist, but the specified URL is within a list (1) or document library, security information is returned from the effective security scope for the specified document location.

The **Individual URL Security Result Set** MUST ONLY be returned if the specified document location is contained within a list (1) or document library. Otherwise, the **NULL Individual URL Security Result Set** (section [3.1.4.57.2](#)) MUST be returned. If returned, the **Individual URL Security Result Set** MUST contain a single row. The **Individual URL Security Result Set** is defined in [\[MS-WSSFO2\]](#) section 2.2.5.10.

3.1.4.57.4 Document Link Information Result Set

The **Document Link Information Result Set** returns information about each forward link from the document and backward link to the document within the site collection. The result set MUST be returned and MUST contain one row for each forward link within the specified document, and one row for each backward link to the document within the specified site collection. The T-SQL syntax for the result set is as follows:

```
LinkDirName          nvarchar(256),
LinkLeafName         nvarchar(128),
LinkType             tinyint,
LinkSecurity         tinyint,
LinkDynamic          tinyint,
LinkServerRel        bit,
LinkStatus           tinyint,
PointsToDir          bit,
WebPartId            int,
LinkNumber           int,
WebId                uniqueidentifier,
Search               ntext,
FieldId              uniqueidentifier;
```

LinkDirName: The directory name of the directory containing the linked object. This value MUST NOT be NULL.

LinkLeafName: The leaf name of the linked object. This value MUST NOT be NULL.

LinkType: Type of the link. Refer to [\[MS-WSSFO2\]](#), section [2.2.3.10](#), for valid values.

LinkSecurity: A 1-byte (**tinyint**) value represented as a single upper case ASCII character specifying the link's security type. Refer to [\[MS-WSSFO2\]](#), section [2.2.3.9](#) for valid values.

LinkDynamic: A 1-byte (**tinyint**) value represented as a single upper case ASCII character that tracks various special link types. The value MUST be one of the values in the following table.

Value	Description
"68" (D)	The URL is "dynamic", which is a link to <Site URL>/_vti_bin/shtml.dll/DirName/LeafName. Such links are used to call the SmartHTML interpreter on a file.
"71" (G)	A non-absolute link from an uncustomized document that does not fall into any other category.

Value	Description
"72" (H)	The URL is a history link; that is, it contains a path segment with the string "_vti_history".
"76" (L)	The URL is to a layouts page; that is, it contains a path segment with the string "_layouts".
"83" (S)	The URL is "static", which is the default, and requires no special handling.

LinkServerRel: A bit flag that specifies whether the link URL is server-relative URL. A value of "1" specifies a server-relative URL. This value MUST be NULL for a backward link.

LinkStatus: The document store type value of the document targeted by a link. This value MUST be "128" for a backward link. If the forward link target is a document that does not exist, or if the forward link refers to a target that exists outside the specified site collection, or if it refers to a location that could not be verified, this value MUST be NULL.

PointsToDir: A bit flag specifying whether the target of the forward link was a directory and has been modified to target a Welcome page. This value MUST be NULL for a backward link. For a forward link, if the target is a directory where a Welcome page is specified, the link MUST be changed to the URL of the Welcome page and this value MUST be "1" so that the link can be distinguished from an explicit link to the Welcome page; otherwise this value MUST be zero ("0").

WebPartId: MUST be NULL.

LinkNumber: MUST be NULL.

WebId: Site identifier for backward links.

Search: Search parameters for backward links.

FieldId: If the link is for a list item field (1) within this document, this is the field identifier of the field (1) to which this the link belongs.

3.1.4.57.5 Document Setup Path Result Set

The T-SQL syntax for the **Document Setup Path Result Set** is as follows:

```
{DocSetupPath}          nvarchar(255);
```

{DocSetupPath}: For a document that is now or once was uncustomized, this contains the setup path fragment relative to the base setup path where the content stream of this document can be found. This value MUST be NULL if the document was never uncustomized.

3.1.4.58 proc_GetListDataLinks

The **proc_GetListDataLinks** stored procedure is called to get the list of dirty field (1) links for a range of list items in a site (2), sorted alphabetically from (**@FirstDirName, @FirstLeafName, @FirstLevel**) to (**@LastDirName, @LastLeafName, @LastLevel**), inclusive of the endpoints. The purpose of this stored procedure is to finish cleaning up links after list (1) location or other metadata has been updated. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetListDataLinks (
```

```

@SiteId                uniqueidentifier,
@WebId                 uniqueidentifier,
@FirstDirName          nvarchar(256),
@FirstLeafName         nvarchar(128),
@FirstLevel            tinyint,
@LastDirName           nvarchar(256),
@LastLeafName          nvarchar(128),
@LastLevel             tinyint,
@GetWebListForNormalization bit,
@RequestGuid           uniqueidentifier = NULL OUTPUT
);

```

@SiteId: The site collection identifier of the site collection which contains the specified list (1).

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

@FirstDirName: First allowable directory name to be returned, in alphabetic order.

@FirstLeafName: First allowable leaf name to be returned, in alphabetic order. This does not have to be a member of the directory **@FirstDirName**.

@FirstLevel: First allowable publishing level to be returned, in numeric order. The level does not have to be the level of a leaf in the directory **@FirstDirName**.

@LastDirName: Last allowable directory name to be returned, in alphabetical order.

@LastLeafName: Last allowable Leaf name to be returned, in alphabetical order. This does not have to be a member of the directory **@LastDirName**.

@LastLevel: Last allowable publishing level to be returned, in numeric order. This level does not have to be the level of a leaf in the directory **@LastDirName**.

@GetWebListForNormalization: Bit flag indicating that **Web List For Normalization Result Set** (section [3.1.4.58.1](#)) MUST be returned.

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

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: Conditionally returns the **Web List For Normalization Result Set** if the **@GetWebListForNormalization** flag is set to 1, and MUST return the **List Data Link Information Result Set** (section [3.1.4.58.2](#)).

3.1.4.58.1 Web List For Normalization Result Set

If the **@GetWebListForNormalization** flag is set to 1, the **Web List For Normalization Result Set** returns a list of URLs for subsites of the specified site collection. This will be an empty result set if there are no subsites in the site collection. The T-SQL syntax for the result set is as follows:

```
FullUrl    nvarchar(256);
```

FullUrl: Full URL of a subsite in the site collection.

3.1.4.58.2 List Data Link Information Result Set

The **List Data Link Information Result Set** returns the list of dirty field (1) links for a range of list items in a site (2), sorted alphabetically from (*@FirstDirName, @FirstLeafName, @FirstLevel*) to (*@LastDirName, @LastLeafName, @LastLevel*), inclusive of the endpoints. This result set returns only rows where **FieldId** is not NULL, and contains zero or more rows. The result set is defined using T-SQL syntax as follows:

```
DirName          nvarchar(256),
LeafName         nvarchar(128),
Level            tinyint,
FieldId          uniqueidentifier,
TargetDirName    nvarchar(256),
TargetLeafName   nvarchar(128),
Type             tinyint,
Security         tinyint,
Dynamic          tinyint,
ServerRel        bit,
Type             tinyint,
PointsToDir      bit;
```

DirName: The directory name of the directory containing the source object of the link. This value MUST NOT be NULL.

LeafName: The leaf name of the source object.

Level: The Publishing Level Type value (see [\[MS-WSSFO2\]](#) section 2.2.2.6) defining the publishing level of the source object.

FieldId: The field identifier of the source object. This value MUST NOT be NULL.

TargetDirName: The directory name of the directory containing the target object of the link. This value MUST NOT be NULL.

TargetLeafName: The leaf name of the linked object. This value MUST NOT be NULL.

Type: Type of the link. The value MUST be one of the values described in the following table.

Value	Description
"A"	The link is from the ACTION attribute of an HTML form tag.
"B"	The link is from the attribute markup of a bot.
"C"	The link is from an auto-generated table of contents. Agents can ignore the link type when determining unreferenced files within a site (2).
"D"	The link references programmatic content, as in the HTML OBJECT or APPLET tags.
"E"	The link is from a cascading style sheet (CSS).
"F"	The link is from the SRC attribute of an HTML FRAME tag.
"G"	The link is to a dynamic Web template for the containing document.
"H"	The link is from an HTML HREF attribute. This can also be used as a default link type value if a more precise type does not apply.

Value	Description
"I"	The link is to a document that the containing document includes via an include bot.
"J"	The link is from a field (1) of this list item.
"K"	Identical to "H", except that the link also specifies an HTML bookmark.
"L"	The link is a target in an HTML image map generated from an image map bot.
"M"	The link is to an image used in an HTML image map generated from an image map bot.
"O"	The link is part of a cross-page URL connection.
"P"	The link is part of the markup of a URL within the source of the containing document.
"Q"	The link references a CSS document that provides style information for the containing document.
"R"	The link is from the master page file attribute of the @Page directive in the containing document.
"S"	The link is from an HTML SRC attribute.
"T"	The link is to the index file used by a text search bot on this Web page.
"V"	The link is based on the properties of the document, rather than anything in the document stream. The link type is used in tracking the link between a site (2) and the master page URL used for the site (2).
"X"	The link is from an XML island within an HTML document.
"Y"	The link references an HTML document whose HTML BODY tag attributes are used as a template for the attributes of the containing document's BODY tag.
"Z"	The link is part of the markup of a URL that exists in a URL zone in the containing document, and is consequently not stored within the source of the containing document.

Security: Type of security for the link. The value **MUST** be one of the values in the following table.

Value	Description
"H"	The link is to an "HTTP:" URL.
"S"	The link is to an "HTTPS:" URL.
"T"	The link is to an "SHTTP:" URL.
"U"	The link transport security is unknown.

Dynamic: A 1-byte (**tinyint**) value represented as a single upper case ASCII character that specifies the special link types. The value **MUST** be one of the values in the following table.

Value	Description
"S"	The URL is static, which is the default, and requires no special handling.
"D"	The URL is dynamic, which is a link to <Site URL>/_vti_bin/shtml.dll/DirName/LeafName. Such links are used to call the SmarHTML interpreter on a file.
"L"	The URL is to a layouts page. In other words, it contains a path segment with the string

Value	Description
	"_layouts".
"H"	The URL is a history link. In other words, it contains a path segment with the string "_vti_history".
"G"	A non-absolute link from an uncustomized document that does not fall into any other category.

ServerRel: A bit flag that specifies whether the link URL is a server-relative URL. A value of "1" specifies a server-relative URL. This value MUST NOT be NULL.

Type: Type of the linked object. This value MUST be one of the values in the following table.

Value	Description
"0"	File
"1"	Directory
"2"	Site (2)
"128"	Backward link

PointsToDir: MUST be "1" if the link is to a directory. Otherwise, it MUST be zero ("0"). This value MUST NOT be NULL.

3.1.4.59 proc_GetListDataSource

The **proc_GetListDataSource** stored procedure is called to retrieve a data source for a list (1). The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetListDataSource (
    @ListId          uniqueidentifier,
    @RequestGuid     uniqueidentifier = NULL OUTPUT
);
```

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

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

Return Code Values: An integer that MUST be zero.

Result Sets: MUST return the **List Data Source Result Set** (section [3.1.4.59.1](#)).

3.1.4.59.1 List Data Source Result Set

The **List Data Source Result Set** returns the data source and **Entity** identifier for the list (1). The **List Data Source Result Set** MUST return one row in the result set. The T-SQL syntax for the result set is as follows:

```
DataSource      nvarchar(max) NOT NULL,
EntityId        int NOT NULL;
```

DataSource: A string that MUST contain the XML fragment for the data source of the list (1). The schema of this fragment is defined by [List Data Source](#).

EntityId: The identifier for the Entity of the list (1).

3.1.4.60 `proc_GetListItemsTreeToDelete`

The `proc_GetListItemsTreeToDelete` stored procedure is called to get information about child items and the parent item of a cascading delete operation. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetListItemsTreeToDelete (  
    @ISiteId        uniqueidentifier,  
    @IWebId         uniqueidentifier,  
    @IListId        uniqueidentifier,  
    @IItemId        int,  
    @MaxLimit       int  
);
```

@ISiteId: The site collection identifier of the site collection containing the parent item of the cascading delete operation.

@IWebId: The site identifier of the site (2) containing the parent item.

@IListId: The list identifier of the list (1) containing the parent item.

@IItemId: The list item identifier of the parent item.

@MaxLimit: An integer that specifies the maximum number of child items allowed to be part of the cascading delete operation.

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

Value	Description
0	Successful execution.
1142	The number of child items found is equal to or greater than @MaxLimit, which is calculated as in List Items Tree Result Set (section 3.1.4.60.1).
8239	All of the following conditions are true for at least one list item, document or folder and at least one relationship lookup field of the list (1) containing it: <ul style="list-style-type: none">▪ The relationship lookup field has the restrict behavior set on it.▪ Either of the following is true:<ul style="list-style-type: none">▪ The target list (1) of the relationship lookup field is the list (1) containing a child item of the cascading delete operation and either of the following is true for its value for the relationship lookup field in its most recent published or draft version:<ul style="list-style-type: none">▪ It is the list item identifier of that child item.▪ It is the list item identifier of a list item, document or folder contained in that child item if it is a folder.▪ The target list (1) of the relationship lookup field is the list (1) containing the parent item of the cascading delete operation and either of the following is true for its value for the

Value	Description
	<p>relationship lookup field in its most recent published or draft version:</p> <ul style="list-style-type: none"> ▪ It is the list item identifier of the parent item. ▪ It is the list item identifier of a list item, document or folder contained in the parent item if it is a folder.

Result sets: MUST return the **List Items Tree Result Set** if execution completes successfully, or the **Restrict Delete Error Result Set** (section [3.1.4.60.2](#)) if the return value is 8239.

3.1.4.60.1 List Items Tree Result Set

The **List Items Tree Delete Result Set** returns information about the set of child items and the parent item of the cascading delete operation. The set of child items MUST be calculated as the union of the following sets:

- All list items, documents or folders in all lists (1) in the site (2) which contain at least one relationship lookup field which satisfies the following conditions:
 - The list item, document or folder is not the parent item of the cascading delete operation.
 - The relationship lookup field has a cascading behavior set on it.
 - The target list (1) of the relationship lookup field is the list (1) containing the parent item of the cascading delete operation and either of the following is true for its value for the relationship lookup field in its most recent published or draft version:
 - It is the list item identifier of the parent item.
 - It is the list item identifier of a list item, document or folder contained in the parent item of the cascading delete operation if it is a folder.
- All list items, documents or folders in all lists (1) in the site (2) which contain at least one relationship lookup field which satisfies the following conditions:
 - The list item, document or folder is not the parent item of the cascading delete operation.
 - The relationship lookup field has a cascading behavior set on it.
 - The target list (1) of the relationship lookup field is the list (1) containing a child item of the cascading delete operation and either of the following is true for its value for the relationship lookup field in its most recent published or draft version:
 - It is the list item identifier of that child item.
 - It is the list item identifier of a list item, document or folder contained in that child item if it is a folder.

```
ListId      uniqueidentifier,
ItemId     int,
DirName    nvarchar (256),
LeafName   nvarchar(128)
```

ListId: The list identifier of the list (1) containing the child item or the parent item of the cascading delete operation.

ItemId: The list item identifier of the child item or the parent item.

DirName: The directory name of the child item or the parent item.

LeafName: The leaf name of the child item or the parent item.

3.1.4.60.2 Restrict Delete Error Result Set

The **Restrict Delete Error Result Set** MUST return information about the list (1) and target list (1) of any one relationship lookup field for any list item, document or folder in the site (2) that satisfies the following conditions:

- The relationship lookup field has the restrict behavior set on it.
- Either of the following is true:
 - The target list (1) of the relationship lookup field is the list (1) containing a child item of the cascading delete operation and either of the following is true for its value for the relationship lookup field in its most recent published or draft version:
 - It is the list item identifier of that child item.
 - It is the list item identifier of a list item, document or folder contained in that child item if it is a folder.
 - The target list (1) of the relationship lookup field is the list (1) containing the parent item of the cascading delete operation and either of the following is true for its value for the relationship lookup field in its most recent published or draft version:
 - It is the list item identifier of the parent item.
 - It is the list item identifier of a list item, document or folder contained in the parent item if it is a folder.

It MUST return one row.

```
{Error_ListId}      uniqueidentifier,  
{Error_LookupListId} uniqueidentifier
```

{Error_ListId}: The list identifier of the list (1) of the relationship lookup field.

{Error_LookupListId}: The list identifier of the target list (1) of the relationship lookup field.

3.1.4.61 proc_GetNewListItemId

The **proc_GetNewListItemId** stored procedure is called to get the next available row identifier of a list (1) and the number of rows that will be copied or moved when copying or moving a URL to that list (1). The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetNewListItemId( @SiteId          uniqueidentifier, @FullUrl  
nvarchar(260), @NewListId      uniqueidentifier, @bIsCopy          bit,  
@NewDoclibRowId int OUTPUT, @MaxNewRows      int OUTPUT
```

);

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

@FullUrl: The full URL to copy or move.

@NewListId: The list identifier of the target list.

@bIsCopy: A bit flag that indicates whether to copy or move the URL specified by **@FullUrl**. The value MUST be 1 to copy or MUST be zero to move.

@NewDocLibRowId: This is an output parameter whose value indicates the next available row identifier in the target list (1) where the URL specified by **@FullUrl** is being copied or moved to. The return value MUST be obtained by calling **proc_GenerateNextId** when the URL specified by **@FullUrl** is a file or folder; otherwise is undefined. See [\[MS-WSSFO2\]](#) section 3.1.5.23 for **proc_GenerateNextId**.

@MaxNewRows: This is an output parameter whose value indicates the total number of items as specified by **@FullUrl** that will be copied or moved. The return value MUST be 1 if the URL specified by **@FullUrl** is a file; MUST be the number of items contained in the folder if the URL specified by **@FullUrl** is a folder; otherwise is undefined.

Return values: An integer that MUST be zero.

3.1.4.62 **proc_GetRelatedFieldsForList**

The **proc_GetRelatedFieldsForList** stored procedure is called to get information about relationship lookup fields in lists in a specified site collection whose target list (1) is a specified list (1). The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetRelatedFieldsForList (  
    @SiteId          uniqueidentifier,  
    @ListId          uniqueidentifier,  
    @RequestGuid     uniqueidentifier = null OUTPUT  
);
```

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

@ListId: The list identifier of the specified list (1).

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

Return values: An integer that MUST be zero.

Result sets: MUST return the **List Related Fields Result Set** (section [3.1.4.62.1](#)).

3.1.4.62.1 **List Related Fields Result Set**

The **List Related fields Result Set** returns information about all the relationship lookup fields in lists (1) in the specified site collection whose target list (1) is the specified list (1). The **List Related Fields Result Set** MUST return one row for each of the relationship lookup fields whose target list (1) is the specified list (1). If there are no such relationship lookup fields, it MUST NOT return any rows. The T-SQL syntax for the result set is as follows:

```
tp_WebId uniqueidentifier NOT NULL,  
tp_Id uniqueidentifier NOT NULL,  
FieldId uniqueidentifier NOT NULL,  
DeleteBehavior tinyint NOT NULL
```

tp_WebId: The site identifier of the site (2) that contains the relationship lookup field.

tp_Id: The list identifier of the list (1) that contains the relationship lookup field.

FieldId: The field identifier of the relationship lookup field.

DeleteBehavior: The relationship delete behavior set on the relationship lookup field. It MUST be set to zero if no relationship delete behavior is set on it. Otherwise it MUST be set to one of values as specified in [Relationship Delete Behavior Type](#).

3.1.4.63 proc_GetSiteDeletionBatch

The **proc_GetSiteDeletionBatch** stored procedure is called to retrieve the list of site collections that are scheduled to be deleted. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetSiteDeletionBatch (  
    @DeletionTime    datetime,  
    @DeletionId      bigint  
);
```

@DeletionTime: the date and time at which the last site collection that was retrieved using this stored procedure was scheduled for deletion.

@DeletionId: An integer that represents the last site collection that was retrieved using this stored procedure. This can be NULL.

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: This procedure MUST return the **Site Deletion Batch Result Set** (section [3.1.4.63.1](#)).

3.1.4.63.1 Site Deletion Batch Result Set

The **Site Deletion Batch Result Set** returns one row for each site collection that has data to delete. It returns zero rows if there are no such site collections and at most 1000 rows. The T-SQL syntax for the result set is as follows.

```
Id                bigint NOT NULL,  
SiteId            uniqueidentifier NOT NULL,  
DeletionTime      datetime NOT NULL;
```

Id: An integer that uniquely represents a site collection that has data scheduled to be deleted.

SiteId: The site collection identifier of the site collection that has data scheduled to be deleted.

DeletionTime: The date and time at which the site collection was scheduled to be deleted.

3.1.4.64 proc_GetUrlDocId

The **proc_GetUrlDocId** stored procedure is called to retrieve the directory name and leaf name of a specified document. The T-SQL syntax for the stored procedure is as follows.

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

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

@WebId: The site identifier of the site (2) that contains the specified document.

@DocId: The document identifier (2) of the specified document.

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

Return values: An integer return code that MUST be included the following table.

Value	Description
0	Successful execution.
2	The specified document does not exist.

Result sets: MUST return 1 **Directory And Leaf Names Result Set** (section [3.1.4.64.1](#)).

3.1.4.64.1 Directory And Leaf Names Result Set

The **Directory And Leaf Names Result Set** contains one row of the directory name and leaf name of the document whose document identifier (2) is @DocId that exists in the site (2) whose site identifier is @WebId that exists in the site collection whose site collection identifier is @SiteId. The **Directory And Leaf Names Result Set** MUST be returned and MUST contain one row if the document exists. The T-SQL syntax for the result set is as follows:

```
DirName    nvarchar(256),  
LeafName   nvarchar(128);
```

DirName: The store-relative form directory name of the document.

LeafName: The leaf name of the document.

3.1.4.65 proc_GetWebUrlFromId

The **proc_GetWebUrlFromId** stored procedure is called to get the full URL of a site (2) from a specified site identifier. The T-SQL syntax for the stored procedure is as follows:

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


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

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

Return values: An integer that MUST be zero.

Result sets: MUST return 1 URL result set as defined in [\[MS-WSSFO2\]](#) section 2.2.5.25.

3.1.4.66 `proc_InsertEventSubscriptionJunctionEntries`

The `proc_InsertEventSubscriptionJunctionEntries` stored procedure is called to insert up to 256 junction entries for an alert (1). The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_InsertEventSubscriptionJunctionEntries (
    @e001      bigint = NULL,
    @s001      uniqueidentifier = NULL,
    @l001      varbinary(max) = NULL,
    @e002      bigint = NULL,
    @s002      uniqueidentifier = NULL,
    @l002      varbinary(max) = NULL,
    @e003      bigint = NULL,
    @s003      uniqueidentifier = NULL,
    @l003      varbinary(max) = NULL,
    @e004      bigint = NULL,
    @s004      uniqueidentifier = NULL,
    @l004      varbinary(max) = NULL,
    @e005      bigint = NULL,
    @s005      uniqueidentifier = NULL,
    @l005      varbinary(max) = NULL,
    @e006      bigint = NULL,
    @s006      uniqueidentifier = NULL,
    @l006      varbinary(max) = NULL,
    @e007      bigint = NULL,
    @s007      uniqueidentifier = NULL,
    @l007      varbinary(max) = NULL,
    @e008      bigint = NULL,
    @s008      uniqueidentifier = NULL,
    @l008      varbinary(max) = NULL,
    @e009      bigint = NULL,
    @s009      uniqueidentifier = NULL,
    @l009      varbinary(max) = NULL,
    @e010      bigint = NULL,
    @s010      uniqueidentifier = NULL,
    @l010      varbinary(max) = NULL,
    @e011      bigint = NULL,
    @s011      uniqueidentifier = NULL,
    @l011      varbinary(max) = NULL,
    @e012      bigint = NULL,
    @s012      uniqueidentifier = NULL,
    @l012      varbinary(max) = NULL,
    @e013      bigint = NULL,
    @s013      uniqueidentifier = NULL,
    @l013      varbinary(max) = NULL,
    @e014      bigint = NULL,
    @s014      uniqueidentifier = NULL,
    @l014      varbinary(max) = NULL,
    @e015      bigint = NULL,
    @s015      uniqueidentifier = NULL,
```

@l015 varbinary(max) = NULL,
@e016 bigint = NULL,
@s016 uniqueidentifier = NULL,
@l016 varbinary(max) = NULL,
@e017 bigint = NULL,
@s017 uniqueidentifier = NULL,
@l017 varbinary(max) = NULL,
@e018 bigint = NULL,
@s018 uniqueidentifier = NULL,
@l018 varbinary(max) = NULL,
@e019 bigint = NULL,
@s019 uniqueidentifier = NULL,
@l019 varbinary(max) = NULL,
@e020 bigint = NULL,
@s020 uniqueidentifier = NULL,
@l020 varbinary(max) = NULL,
@e021 bigint = NULL,
@s021 uniqueidentifier = NULL,
@l021 varbinary(max) = NULL,
@e022 bigint = NULL,
@s022 uniqueidentifier = NULL,
@l022 varbinary(max) = NULL,
@e023 bigint = NULL,
@s023 uniqueidentifier = NULL,
@l023 varbinary(max) = NULL,
@e024 bigint = NULL,
@s024 uniqueidentifier = NULL,
@l024 varbinary(max) = NULL,
@e025 bigint = NULL,
@s025 uniqueidentifier = NULL,
@l025 varbinary(max) = NULL,
@e026 bigint = NULL,
@s026 uniqueidentifier = NULL,
@l026 varbinary(max) = NULL,
@e027 bigint = NULL,
@s027 uniqueidentifier = NULL,
@l027 varbinary(max) = NULL,
@e028 bigint = NULL,
@s028 uniqueidentifier = NULL,
@l028 varbinary(max) = NULL,
@e029 bigint = NULL,
@s029 uniqueidentifier = NULL,
@l029 varbinary(max) = NULL,
@e030 bigint = NULL,
@s030 uniqueidentifier = NULL,
@l030 varbinary(max) = NULL,
@e031 bigint = NULL,
@s031 uniqueidentifier = NULL,
@l031 varbinary(max) = NULL,
@e032 bigint = NULL,
@s032 uniqueidentifier = NULL,
@l032 varbinary(max) = NULL,
@e033 bigint = NULL,
@s033 uniqueidentifier = NULL,
@l033 varbinary(max) = NULL,
@e034 bigint = NULL,
@s034 uniqueidentifier = NULL,
@l034 varbinary(max) = NULL,
@e035 bigint = NULL,

@s035 uniqueidentifier = NULL,
@l035 varbinary(max) = NULL,
@e036 bigint = NULL,
@s036 uniqueidentifier = NULL,
@l036 varbinary(max) = NULL,
@e037 bigint = NULL,
@s037 uniqueidentifier = NULL,
@l037 varbinary(max) = NULL,
@e038 bigint = NULL,
@s038 uniqueidentifier = NULL,
@l038 varbinary(max) = NULL,
@e039 bigint = NULL,
@s039 uniqueidentifier = NULL,
@l039 varbinary(max) = NULL,
@e040 bigint = NULL,
@s040 uniqueidentifier = NULL,
@l040 varbinary(max) = NULL,
@e041 bigint = NULL,
@s041 uniqueidentifier = NULL,
@l041 varbinary(max) = NULL,
@e042 bigint = NULL,
@s042 uniqueidentifier = NULL,
@l042 varbinary(max) = NULL,
@e043 bigint = NULL,
@s043 uniqueidentifier = NULL,
@l043 varbinary(max) = NULL,
@e044 bigint = NULL,
@s044 uniqueidentifier = NULL,
@l044 varbinary(max) = NULL,
@e045 bigint = NULL,
@s045 uniqueidentifier = NULL,
@l045 varbinary(max) = NULL,
@e046 bigint = NULL,
@s046 uniqueidentifier = NULL,
@l046 varbinary(max) = NULL,
@e047 bigint = NULL,
@s047 uniqueidentifier = NULL,
@l047 varbinary(max) = NULL,
@e048 bigint = NULL,
@s048 uniqueidentifier = NULL,
@l048 varbinary(max) = NULL,
@e049 bigint = NULL,
@s049 uniqueidentifier = NULL,
@l049 varbinary(max) = NULL,
@e050 bigint = NULL,
@s050 uniqueidentifier = NULL,
@l050 varbinary(max) = NULL,
@e051 bigint = NULL,
@s051 uniqueidentifier = NULL,
@l051 varbinary(max) = NULL,
@e052 bigint = NULL,
@s052 uniqueidentifier = NULL,
@l052 varbinary(max) = NULL,
@e053 bigint = NULL,
@s053 uniqueidentifier = NULL,
@l053 varbinary(max) = NULL,
@e054 bigint = NULL,
@s054 uniqueidentifier = NULL,
@l054 varbinary(max) = NULL,

@e055 bigint = NULL,
@s055 uniqueidentifier = NULL,
@l055 varbinary(max) = NULL,
@e056 bigint = NULL,
@s056 uniqueidentifier = NULL,
@l056 varbinary(max) = NULL,
@e057 bigint = NULL,
@s057 uniqueidentifier = NULL,
@l057 varbinary(max) = NULL,
@e058 bigint = NULL,
@s058 uniqueidentifier = NULL,
@l058 varbinary(max) = NULL,
@e059 bigint = NULL,
@s059 uniqueidentifier = NULL,
@l059 varbinary(max) = NULL,
@e060 bigint = NULL,
@s060 uniqueidentifier = NULL,
@l060 varbinary(max) = NULL,
@e061 bigint = NULL,
@s061 uniqueidentifier = NULL,
@l061 varbinary(max) = NULL,
@e062 bigint = NULL,
@s062 uniqueidentifier = NULL,
@l062 varbinary(max) = NULL,
@e063 bigint = NULL,
@s063 uniqueidentifier = NULL,
@l063 varbinary(max) = NULL,
@e064 bigint = NULL,
@s064 uniqueidentifier = NULL,
@l064 varbinary(max) = NULL,
@e065 bigint = NULL,
@s065 uniqueidentifier = NULL,
@l065 varbinary(max) = NULL,
@e066 bigint = NULL,
@s066 uniqueidentifier = NULL,
@l066 varbinary(max) = NULL,
@e067 bigint = NULL,
@s067 uniqueidentifier = NULL,
@l067 varbinary(max) = NULL,
@e068 bigint = NULL,
@s068 uniqueidentifier = NULL,
@l068 varbinary(max) = NULL,
@e069 bigint = NULL,
@s069 uniqueidentifier = NULL,
@l069 varbinary(max) = NULL,
@e070 bigint = NULL,
@s070 uniqueidentifier = NULL,
@l070 varbinary(max) = NULL,
@e071 bigint = NULL,
@s071 uniqueidentifier = NULL,
@l071 varbinary(max) = NULL,
@e072 bigint = NULL,
@s072 uniqueidentifier = NULL,
@l072 varbinary(max) = NULL,
@e073 bigint = NULL,
@s073 uniqueidentifier = NULL,
@l073 varbinary(max) = NULL,
@e074 bigint = NULL,
@s074 uniqueidentifier = NULL,

@l074 varbinary(max) = NULL,
@e075 bigint = NULL,
@s075 uniqueidentifier = NULL,
@l075 varbinary(max) = NULL,
@e076 bigint = NULL,
@s076 uniqueidentifier = NULL,
@l076 varbinary(max) = NULL,
@e077 bigint = NULL,
@s077 uniqueidentifier = NULL,
@l077 varbinary(max) = NULL,
@e078 bigint = NULL,
@s078 uniqueidentifier = NULL,
@l078 varbinary(max) = NULL,
@e079 bigint = NULL,
@s079 uniqueidentifier = NULL,
@l079 varbinary(max) = NULL,
@e080 bigint = NULL,
@s080 uniqueidentifier = NULL,
@l080 varbinary(max) = NULL,
@e081 bigint = NULL,
@s081 uniqueidentifier = NULL,
@l081 varbinary(max) = NULL,
@e082 bigint = NULL,
@s082 uniqueidentifier = NULL,
@l082 varbinary(max) = NULL,
@e083 bigint = NULL,
@s083 uniqueidentifier = NULL,
@l083 varbinary(max) = NULL,
@e084 bigint = NULL,
@s084 uniqueidentifier = NULL,
@l084 varbinary(max) = NULL,
@e085 bigint = NULL,
@s085 uniqueidentifier = NULL,
@l085 varbinary(max) = NULL,
@e086 bigint = NULL,
@s086 uniqueidentifier = NULL,
@l086 varbinary(max) = NULL,
@e087 bigint = NULL,
@s087 uniqueidentifier = NULL,
@l087 varbinary(max) = NULL,
@e088 bigint = NULL,
@s088 uniqueidentifier = NULL,
@l088 varbinary(max) = NULL,
@e089 bigint = NULL,
@s089 uniqueidentifier = NULL,
@l089 varbinary(max) = NULL,
@e090 bigint = NULL,
@s090 uniqueidentifier = NULL,
@l090 varbinary(max) = NULL,
@e091 bigint = NULL,
@s091 uniqueidentifier = NULL,
@l091 varbinary(max) = NULL,
@e092 bigint = NULL,
@s092 uniqueidentifier = NULL,
@l092 varbinary(max) = NULL,
@e093 bigint = NULL,
@s093 uniqueidentifier = NULL,
@l093 varbinary(max) = NULL,
@e094 bigint = NULL,

@s094 uniqueidentifier = NULL,
@l094 varbinary(max) = NULL,
@e095 bigint = NULL,
@s095 uniqueidentifier = NULL,
@l095 varbinary(max) = NULL,
@e096 bigint = NULL,
@s096 uniqueidentifier = NULL,
@l096 varbinary(max) = NULL,
@e097 bigint = NULL,
@s097 uniqueidentifier = NULL,
@l097 varbinary(max) = NULL,
@e098 bigint = NULL,
@s098 uniqueidentifier = NULL,
@l098 varbinary(max) = NULL,
@e099 bigint = NULL,
@s099 uniqueidentifier = NULL,
@l099 varbinary(max) = NULL,
@e100 bigint = NULL,
@s100 uniqueidentifier = NULL,
@l100 varbinary(max) = NULL,
@e101 bigint = NULL,
@s101 uniqueidentifier = NULL,
@l101 varbinary(max) = NULL,
@e102 bigint = NULL,
@s102 uniqueidentifier = NULL,
@l102 varbinary(max) = NULL,
@e103 bigint = NULL,
@s103 uniqueidentifier = NULL,
@l103 varbinary(max) = NULL,
@e104 bigint = NULL,
@s104 uniqueidentifier = NULL,
@l104 varbinary(max) = NULL,
@e105 bigint = NULL,
@s105 uniqueidentifier = NULL,
@l105 varbinary(max) = NULL,
@e106 bigint = NULL,
@s106 uniqueidentifier = NULL,
@l106 varbinary(max) = NULL,
@e107 bigint = NULL,
@s107 uniqueidentifier = NULL,
@l107 varbinary(max) = NULL,
@e108 bigint = NULL,
@s108 uniqueidentifier = NULL,
@l108 varbinary(max) = NULL,
@e109 bigint = NULL,
@s109 uniqueidentifier = NULL,
@l109 varbinary(max) = NULL,
@e110 bigint = NULL,
@s110 uniqueidentifier = NULL,
@l110 varbinary(max) = NULL,
@e111 bigint = NULL,
@s111 uniqueidentifier = NULL,
@l111 varbinary(max) = NULL,
@e112 bigint = NULL,
@s112 uniqueidentifier = NULL,
@l112 varbinary(max) = NULL,
@e113 bigint = NULL,
@s113 uniqueidentifier = NULL,
@l113 varbinary(max) = NULL,

@e114 bigint = NULL,
@s114 uniqueidentifier = NULL,
@l114 varbinary(max) = NULL,
@e115 bigint = NULL,
@s115 uniqueidentifier = NULL,
@l115 varbinary(max) = NULL,
@e116 bigint = NULL,
@s116 uniqueidentifier = NULL,
@l116 varbinary(max) = NULL,
@e117 bigint = NULL,
@s117 uniqueidentifier = NULL,
@l117 varbinary(max) = NULL,
@e118 bigint = NULL,
@s118 uniqueidentifier = NULL,
@l118 varbinary(max) = NULL,
@e119 bigint = NULL,
@s119 uniqueidentifier = NULL,
@l119 varbinary(max) = NULL,
@e120 bigint = NULL,
@s120 uniqueidentifier = NULL,
@l120 varbinary(max) = NULL,
@e121 bigint = NULL,
@s121 uniqueidentifier = NULL,
@l121 varbinary(max) = NULL,
@e122 bigint = NULL,
@s122 uniqueidentifier = NULL,
@l122 varbinary(max) = NULL,
@e123 bigint = NULL,
@s123 uniqueidentifier = NULL,
@l123 varbinary(max) = NULL,
@e124 bigint = NULL,
@s124 uniqueidentifier = NULL,
@l124 varbinary(max) = NULL,
@e125 bigint = NULL,
@s125 uniqueidentifier = NULL,
@l125 varbinary(max) = NULL,
@e126 bigint = NULL,
@s126 uniqueidentifier = NULL,
@l126 varbinary(max) = NULL,
@e127 bigint = NULL,
@s127 uniqueidentifier = NULL,
@l127 varbinary(max) = NULL,
@e128 bigint = NULL,
@s128 uniqueidentifier = NULL,
@l128 varbinary(max) = NULL,
@e129 bigint = NULL,
@s129 uniqueidentifier = NULL,
@l129 varbinary(max) = NULL,
@e130 bigint = NULL,
@s130 uniqueidentifier = NULL,
@l130 varbinary(max) = NULL,
@e131 bigint = NULL,
@s131 uniqueidentifier = NULL,
@l131 varbinary(max) = NULL,
@e132 bigint = NULL,
@s132 uniqueidentifier = NULL,
@l132 varbinary(max) = NULL,
@e133 bigint = NULL,
@s133 uniqueidentifier = NULL,

@l133 varbinary(max) = NULL,
@e134 bigint = NULL,
@s134 uniqueidentifier = NULL,
@l134 varbinary(max) = NULL,
@e135 bigint = NULL,
@s135 uniqueidentifier = NULL,
@l135 varbinary(max) = NULL,
@e136 bigint = NULL,
@s136 uniqueidentifier = NULL,
@l136 varbinary(max) = NULL,
@e137 bigint = NULL,
@s137 uniqueidentifier = NULL,
@l137 varbinary(max) = NULL,
@e138 bigint = NULL,
@s138 uniqueidentifier = NULL,
@l138 varbinary(max) = NULL,
@e139 bigint = NULL,
@s139 uniqueidentifier = NULL,
@l139 varbinary(max) = NULL,
@e140 bigint = NULL,
@s140 uniqueidentifier = NULL,
@l140 varbinary(max) = NULL,
@e141 bigint = NULL,
@s141 uniqueidentifier = NULL,
@l141 varbinary(max) = NULL,
@e142 bigint = NULL,
@s142 uniqueidentifier = NULL,
@l142 varbinary(max) = NULL,
@e143 bigint = NULL,
@s143 uniqueidentifier = NULL,
@l143 varbinary(max) = NULL,
@e144 bigint = NULL,
@s144 uniqueidentifier = NULL,
@l144 varbinary(max) = NULL,
@e145 bigint = NULL,
@s145 uniqueidentifier = NULL,
@l145 varbinary(max) = NULL,
@e146 bigint = NULL,
@s146 uniqueidentifier = NULL,
@l146 varbinary(max) = NULL,
@e147 bigint = NULL,
@s147 uniqueidentifier = NULL,
@l147 varbinary(max) = NULL,
@e148 bigint = NULL,
@s148 uniqueidentifier = NULL,
@l148 varbinary(max) = NULL,
@e149 bigint = NULL,
@s149 uniqueidentifier = NULL,
@l149 varbinary(max) = NULL,
@e150 bigint = NULL,
@s150 uniqueidentifier = NULL,
@l150 varbinary(max) = NULL,
@e151 bigint = NULL,
@s151 uniqueidentifier = NULL,
@l151 varbinary(max) = NULL,
@e152 bigint = NULL,
@s152 uniqueidentifier = NULL,
@l152 varbinary(max) = NULL,
@e153 bigint = NULL,

@s153 uniqueidentifier = NULL,
@l153 varbinary(max) = NULL,
@e154 bigint = NULL,
@s154 uniqueidentifier = NULL,
@l154 varbinary(max) = NULL,
@e155 bigint = NULL,
@s155 uniqueidentifier = NULL,
@l155 varbinary(max) = NULL,
@e156 bigint = NULL,
@s156 uniqueidentifier = NULL,
@l156 varbinary(max) = NULL,
@e157 bigint = NULL,
@s157 uniqueidentifier = NULL,
@l157 varbinary(max) = NULL,
@e158 bigint = NULL,
@s158 uniqueidentifier = NULL,
@l158 varbinary(max) = NULL,
@e159 bigint = NULL,
@s159 uniqueidentifier = NULL,
@l159 varbinary(max) = NULL,
@e160 bigint = NULL,
@s160 uniqueidentifier = NULL,
@l160 varbinary(max) = NULL,
@e161 bigint = NULL,
@s161 uniqueidentifier = NULL,
@l161 varbinary(max) = NULL,
@e162 bigint = NULL,
@s162 uniqueidentifier = NULL,
@l162 varbinary(max) = NULL,
@e163 bigint = NULL,
@s163 uniqueidentifier = NULL,
@l163 varbinary(max) = NULL,
@e164 bigint = NULL,
@s164 uniqueidentifier = NULL,
@l164 varbinary(max) = NULL,
@e165 bigint = NULL,
@s165 uniqueidentifier = NULL,
@l165 varbinary(max) = NULL,
@e166 bigint = NULL,
@s166 uniqueidentifier = NULL,
@l166 varbinary(max) = NULL,
@e167 bigint = NULL,
@s167 uniqueidentifier = NULL,
@l167 varbinary(max) = NULL,
@e168 bigint = NULL,
@s168 uniqueidentifier = NULL,
@l168 varbinary(max) = NULL,
@e169 bigint = NULL,
@s169 uniqueidentifier = NULL,
@l169 varbinary(max) = NULL,
@e170 bigint = NULL,
@s170 uniqueidentifier = NULL,
@l170 varbinary(max) = NULL,
@e171 bigint = NULL,
@s171 uniqueidentifier = NULL,
@l171 varbinary(max) = NULL,
@e172 bigint = NULL,
@s172 uniqueidentifier = NULL,
@l172 varbinary(max) = NULL,

@e173 bigint = NULL,
@s173 uniqueidentifier = NULL,
@l173 varbinary(max) = NULL,
@e174 bigint = NULL,
@s174 uniqueidentifier = NULL,
@l174 varbinary(max) = NULL,
@e175 bigint = NULL,
@s175 uniqueidentifier = NULL,
@l175 varbinary(max) = NULL,
@e176 bigint = NULL,
@s176 uniqueidentifier = NULL,
@l176 varbinary(max) = NULL,
@e177 bigint = NULL,
@s177 uniqueidentifier = NULL,
@l177 varbinary(max) = NULL,
@e178 bigint = NULL,
@s178 uniqueidentifier = NULL,
@l178 varbinary(max) = NULL,
@e179 bigint = NULL,
@s179 uniqueidentifier = NULL,
@l179 varbinary(max) = NULL,
@e180 bigint = NULL,
@s180 uniqueidentifier = NULL,
@l180 varbinary(max) = NULL,
@e181 bigint = NULL,
@s181 uniqueidentifier = NULL,
@l181 varbinary(max) = NULL,
@e182 bigint = NULL,
@s182 uniqueidentifier = NULL,
@l182 varbinary(max) = NULL,
@e183 bigint = NULL,
@s183 uniqueidentifier = NULL,
@l183 varbinary(max) = NULL,
@e184 bigint = NULL,
@s184 uniqueidentifier = NULL,
@l184 varbinary(max) = NULL,
@e185 bigint = NULL,
@s185 uniqueidentifier = NULL,
@l185 varbinary(max) = NULL,
@e186 bigint = NULL,
@s186 uniqueidentifier = NULL,
@l186 varbinary(max) = NULL,
@e187 bigint = NULL,
@s187 uniqueidentifier = NULL,
@l187 varbinary(max) = NULL,
@e188 bigint = NULL,
@s188 uniqueidentifier = NULL,
@l188 varbinary(max) = NULL,
@e189 bigint = NULL,
@s189 uniqueidentifier = NULL,
@l189 varbinary(max) = NULL,
@e190 bigint = NULL,
@s190 uniqueidentifier = NULL,
@l190 varbinary(max) = NULL,
@e191 bigint = NULL,
@s191 uniqueidentifier = NULL,
@l191 varbinary(max) = NULL,
@e192 bigint = NULL,
@s192 uniqueidentifier = NULL,

@l192 varbinary(max) = NULL,
@e193 bigint = NULL,
@s193 uniqueidentifier = NULL,
@l193 varbinary(max) = NULL,
@e194 bigint = NULL,
@s194 uniqueidentifier = NULL,
@l194 varbinary(max) = NULL,
@e195 bigint = NULL,
@s195 uniqueidentifier = NULL,
@l195 varbinary(max) = NULL,
@e196 bigint = NULL,
@s196 uniqueidentifier = NULL,
@l196 varbinary(max) = NULL,
@e197 bigint = NULL,
@s197 uniqueidentifier = NULL,
@l197 varbinary(max) = NULL,
@e198 bigint = NULL,
@s198 uniqueidentifier = NULL,
@l198 varbinary(max) = NULL,
@e199 bigint = NULL,
@s199 uniqueidentifier = NULL,
@l199 varbinary(max) = NULL,
@e200 bigint = NULL,
@s200 uniqueidentifier = NULL,
@l200 varbinary(max) = NULL,
@e201 bigint = NULL,
@s201 uniqueidentifier = NULL,
@l201 varbinary(max) = NULL,
@e202 bigint = NULL,
@s202 uniqueidentifier = NULL,
@l202 varbinary(max) = NULL,
@e203 bigint = NULL,
@s203 uniqueidentifier = NULL,
@l203 varbinary(max) = NULL,
@e204 bigint = NULL,
@s204 uniqueidentifier = NULL,
@l204 varbinary(max) = NULL,
@e205 bigint = NULL,
@s205 uniqueidentifier = NULL,
@l205 varbinary(max) = NULL,
@e206 bigint = NULL,
@s206 uniqueidentifier = NULL,
@l206 varbinary(max) = NULL,
@e207 bigint = NULL,
@s207 uniqueidentifier = NULL,
@l207 varbinary(max) = NULL,
@e208 bigint = NULL,
@s208 uniqueidentifier = NULL,
@l208 varbinary(max) = NULL,
@e209 bigint = NULL,
@s209 uniqueidentifier = NULL,
@l209 varbinary(max) = NULL,
@e210 bigint = NULL,
@s210 uniqueidentifier = NULL,
@l210 varbinary(max) = NULL,
@e211 bigint = NULL,
@s211 uniqueidentifier = NULL,
@l211 varbinary(max) = NULL,
@e212 bigint = NULL,

@s212 uniqueidentifier = NULL,
@l212 varbinary(max) = NULL,
@e213 bigint = NULL,
@s213 uniqueidentifier = NULL,
@l213 varbinary(max) = NULL,
@e214 bigint = NULL,
@s214 uniqueidentifier = NULL,
@l214 varbinary(max) = NULL,
@e215 bigint = NULL,
@s215 uniqueidentifier = NULL,
@l215 varbinary(max) = NULL,
@e216 bigint = NULL,
@s216 uniqueidentifier = NULL,
@l216 varbinary(max) = NULL,
@e217 bigint = NULL,
@s217 uniqueidentifier = NULL,
@l217 varbinary(max) = NULL,
@e218 bigint = NULL,
@s218 uniqueidentifier = NULL,
@l218 varbinary(max) = NULL,
@e219 bigint = NULL,
@s219 uniqueidentifier = NULL,
@l219 varbinary(max) = NULL,
@e220 bigint = NULL,
@s220 uniqueidentifier = NULL,
@l220 varbinary(max) = NULL,
@e221 bigint = NULL,
@s221 uniqueidentifier = NULL,
@l221 varbinary(max) = NULL,
@e222 bigint = NULL,
@s222 uniqueidentifier = NULL,
@l222 varbinary(max) = NULL,
@e223 bigint = NULL,
@s223 uniqueidentifier = NULL,
@l223 varbinary(max) = NULL,
@e224 bigint = NULL,
@s224 uniqueidentifier = NULL,
@l224 varbinary(max) = NULL,
@e225 bigint = NULL,
@s225 uniqueidentifier = NULL,
@l225 varbinary(max) = NULL,
@e226 bigint = NULL,
@s226 uniqueidentifier = NULL,
@l226 varbinary(max) = NULL,
@e227 bigint = NULL,
@s227 uniqueidentifier = NULL,
@l227 varbinary(max) = NULL,
@e228 bigint = NULL,
@s228 uniqueidentifier = NULL,
@l228 varbinary(max) = NULL,
@e229 bigint = NULL,
@s229 uniqueidentifier = NULL,
@l229 varbinary(max) = NULL,
@e230 bigint = NULL,
@s230 uniqueidentifier = NULL,
@l230 varbinary(max) = NULL,
@e231 bigint = NULL,
@s231 uniqueidentifier = NULL,
@l231 varbinary(max) = NULL,

@e232 bigint = NULL,
@s232 uniqueidentifier = NULL,
@l232 varbinary(max) = NULL,
@e233 bigint = NULL,
@s233 uniqueidentifier = NULL,
@l233 varbinary(max) = NULL,
@e234 bigint = NULL,
@s234 uniqueidentifier = NULL,
@l234 varbinary(max) = NULL,
@e235 bigint = NULL,
@s235 uniqueidentifier = NULL,
@l235 varbinary(max) = NULL,
@e236 bigint = NULL,
@s236 uniqueidentifier = NULL,
@l236 varbinary(max) = NULL,
@e237 bigint = NULL,
@s237 uniqueidentifier = NULL,
@l237 varbinary(max) = NULL,
@e238 bigint = NULL,
@s238 uniqueidentifier = NULL,
@l238 varbinary(max) = NULL,
@e239 bigint = NULL,
@s239 uniqueidentifier = NULL,
@l239 varbinary(max) = NULL,
@e240 bigint = NULL,
@s240 uniqueidentifier = NULL,
@l240 varbinary(max) = NULL,
@e241 bigint = NULL,
@s241 uniqueidentifier = NULL,
@l241 varbinary(max) = NULL,
@e242 bigint = NULL,
@s242 uniqueidentifier = NULL,
@l242 varbinary(max) = NULL,
@e243 bigint = NULL,
@s243 uniqueidentifier = NULL,
@l243 varbinary(max) = NULL,
@e244 bigint = NULL,
@s244 uniqueidentifier = NULL,
@l244 varbinary(max) = NULL,
@e245 bigint = NULL,
@s245 uniqueidentifier = NULL,
@l245 varbinary(max) = NULL,
@e246 bigint = NULL,
@s246 uniqueidentifier = NULL,
@l246 varbinary(max) = NULL,
@e247 bigint = NULL,
@s247 uniqueidentifier = NULL,
@l247 varbinary(max) = NULL,
@e248 bigint = NULL,
@s248 uniqueidentifier = NULL,
@l248 varbinary(max) = NULL,
@e249 bigint = NULL,
@s249 uniqueidentifier = NULL,
@l249 varbinary(max) = NULL,
@e250 bigint = NULL,
@s250 uniqueidentifier = NULL,
@l250 varbinary(max) = NULL,
@e251 bigint = NULL,
@s251 uniqueidentifier = NULL,

```

@l251      varbinary(max) = NULL,
@e252      bigint = NULL,
@s252      uniqueidentifier = NULL,
@l252      varbinary(max) = NULL,
@e253      bigint = NULL,
@s253      uniqueidentifier = NULL,
@l253      varbinary(max) = NULL,
@e254      bigint = NULL,
@s254      uniqueidentifier = NULL,
@l254      varbinary(max) = NULL,
@e255      bigint = NULL,
@s255      uniqueidentifier = NULL,
@l255      varbinary(max) = NULL,
@e256      bigint = NULL,
@s256      uniqueidentifier = NULL,
@l256      varbinary(max) = NULL,
@RequestGuid uniqueidentifier = NULL OUTPUT
);

```

@e### The event identifier corresponding to a unique event (1) for which subscription data has to be inserted.

@s###: The subscription identifier for which subscription data has to be inserted. If corresponding @ennn is not NULL then @snnn MUST not be NULL.

@l###: Contains implementation-specific data that holds the permissions the user has for lookup fields.

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

Return Code Values: An integer that MUST be zero.

Result Sets: MUST NOT return any result sets.

3.1.4.67 **proc_InsertItemIntoNameValuePair**

The **proc_InsertItemIntoNameValuePair** stored procedure is called to insert indexed fields and their values for the specified list item. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_InsertItemIntoNameValuePair(
@SiteId          uniqueidentifier,
@WebId           uniqueidentifier,
@ListId         uniqueidentifier,
@ItemId         int,
@Level          tinyint = 1,
@FieldId1       uniqueidentifier = NULL,
@FieldValue1    sql_variant = NULL,
@FieldId2       uniqueidentifier = NULL,
@FieldValue2    sql_variant = NULL,
@FieldId3       uniqueidentifier = NULL,
@FieldValue3    sql_variant = NULL,
@FieldId4       uniqueidentifier = NULL,
@FieldValue4    sql_variant = NULL,
@FieldId5       uniqueidentifier = NULL,
@FieldValue5    sql_variant = NULL,
@FieldId6       uniqueidentifier = NULL,
@FieldValue6    sql_variant = NULL,

```

```

@FieldId7                uniqueidentifier = NULL,
@FieldValue7             sql_variant = NULL,
@FieldId8                uniqueidentifier = NULL,
@FieldValue8             sql_variant = NULL,
@FieldId9                uniqueidentifier = NULL,
@FieldValue9             sql_variant = NULL,
@FieldId10              uniqueidentifier = NULL,
@FieldValue10            sql_variant = NULL,
@SelectFromUserData     bit = 0,
@RequestGuid             uniqueidentifier = NULL OUTPUT
);

```

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

@WebId: The site identifier of the site (2) which contains the specified list item.

@ListId: The list identifier of the list (1) which contains the specified list item.

@ItemId: The identifier of the specified list item in the list (1).

@Level: The publishing level. The default value is 1.

@FieldId#: The field identifier of the indexed fields. There are ten FieldId parameters numbered from 1 to 10. The default values are NULL.

@FieldValue#: The value of the indexed fields. There are ten FieldValue parameters numbered from 1 to 10. The default values are NULL.

@SelectFromUserData: An input parameter. If it is set to 1 and @FieldId# is not NULL and @FieldId# is one of the field identifier values in the following table, then associated @FieldValue# is replaced with the corresponding column value of the row in the [AllUserData](#) specified by @ListId, @ItemId and @Level. The default value is zero. This parameter MUST NOT be NULL.

Field Id	Field name(column in the AllUserData table)
1df5e554-ec7e-46a6-901d-d85a3881cb18	tp_Author
d31655d1-1d5b-4511-95a1-7a09e9b75bf2	tp_Editor
8c06beca-0777-48f7-91c7-6da68bc07b69	tp_Created
28cf69c5-fa48-462a-b5cd-27b6f9d2bd5f	tp_Modified
26d0756c-986a-48a7-af35-bf18ab85ff4a	tp_HasCopyDestinations
7841bf41-43d0-4434-9f50-a673baef7631	tp_UIVersion
03e45e84-1992-4d42-9116-26f756012634	tp_ContentTypeId
3881510a-4e4a-4ee8-b102-8ee8e2d0dd4b	tp_CheckoutUserId

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

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

Value	Description
0	Successful execution.
87	The input parameters are incorrect.

Result sets: MUST NOT return any result sets.

3.1.4.68 `proc_InsertItemIntoNameValuePairCollated`

The `proc_InsertItemIntoNameValuePairCollated` stored procedure is called to insert indexed fields and their values for the specified list item for a specific **collation**. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_InsertItemIntoNameValuePairCollated(
    @SiteId          uniqueidentifier,
    @WebId           uniqueidentifier,
    @ListId          uniqueidentifier,
    @ItemId          int,
    @Collation       smallint,
    @Level           tinyint = 1,
    @FieldId1       uniqueidentifier = NULL,
    @FieldValue1    nvarchar(255) = NULL,
    @FieldId2       uniqueidentifier = NULL,
    @FieldValue2    nvarchar(255) = NULL,
    @FieldId3       uniqueidentifier = NULL,
    @FieldValue3    nvarchar(255) = NULL,
    @FieldId4       uniqueidentifier = NULL,
    @FieldValue4    nvarchar(255) = NULL,
    @FieldId5       uniqueidentifier = NULL,
    @FieldValue5    nvarchar(255) = NULL,
    @FieldId6       uniqueidentifier = NULL,
    @FieldValue6    nvarchar(255) = NULL,
    @FieldId7       uniqueidentifier = NULL,
    @FieldValue7    nvarchar(255) = NULL,
    @FieldId8       uniqueidentifier = NULL,
    @FieldValue8    nvarchar(255) = NULL,
    @FieldId9       uniqueidentifier = NULL,
    @FieldValue9    nvarchar(255) = NULL,
    @FieldId10      uniqueidentifier = NULL,
    @FieldValue10   nvarchar(255) = NULL
);

```

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

@WebId: The site identifier of the site (2) which contains the specified list item.

@ListID: The list identifier of the list (1) which contains the specified list item.

@ItemId: The identifier of the specified list item in the list (1).

@Collation: The **collation identifier** of the collation for the specified list item in the list (1).

@Level: The publishing level. The default value is 1.

@FieldId#: The field identifiers of the indexed fields. There are ten FieldId parameters numbered from 1 to 10. The default values are NULL.

@FieldValue#: The value of the indexed fields. There are ten FieldValue parameters numbered from 1 to 10. The default values are NULL.

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

Value	Description
0	Successful execution.
87	The input parameters are not valid.

Result sets: MUST NOT return any result sets.

3.1.4.69 proc_InsertJunction

The **proc_InsertJunction** stored procedure is called to add a value to the set of values of a multivalued lookup field of a specified list item in a list (1).

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

```
PROCEDURE proc_InsertJunction(  
    @SiteId                uniqueidentifier,  
    @DirName               nvarchar(256),  
    @LeafName              nvarchar(128),  
    @FieldId               uniqueidentifier,  
    @Id                    int,  
    @Ordinal               int,  
    @Level                 tinyint = 1,  
    @UIVersion             int = 512,  
    @IsCurrentVersion      bit = 1,  
    @CalculatedVersion     int = 0,  
    @DeleteTransactionId   varbinary(16)= 0x  
    @RequestGuid           uniqueidentifier = NULL OUTPUT  
);
```

@SiteId: The site collection identifier of the site collection which contains the specified list (1) which contains the specified multivalued lookup field.

@DirName: The directory name of the specified list item.

@LeafName: The leaf name of the specified list item.

@FieldId: The field identifier of the specified multivalued lookup field.

@Id: The row identifier of a list item in the list (1) being looked up by the specified multivalued lookup field.

@Ordinal: It MUST be a 0-based ordinal of the row which contains the column corresponding to the specified lookup field. Additional rows are used when a list (1) has more user-defined columns of one or more data types than can fit in a single row of this view.

@Level: A publishing level value specifying the publish status of this list item.

@UIVersion: A user interface (UI) version number associated with the list item.

@IsCurrentVersion: This parameter MUST be 1.

@CalculatedVersion: This parameter MUST be zero.

@DeleteTransactionId: This parameter MUST be 0x.

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

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

Value	Description
0	Successful execution.
2	Insertion Error. @SiteId, @DeleteTransactionId, @IsCurrentVersion, @FieldId, @CalculatedVersion, @Level, @Ordinal do not form a unique entry in AllUserDataJunctions.

Result sets: MUST NOT return any result sets.

3.1.4.70 **proc_InsertListUniqueField**

The **proc_InsertListUniqueField** stored procedure is called to set the unique column constraint on a field (1) in a list (1). The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_InsertListUniqueField (  
    @SiteId      uniqueidentifier,  
    @ListId      uniqueidentifier,  
    @FieldId     uniqueidentifier  
);
```

@SiteId: The site collection identifier of the site collection which contains the specified list (1).

@ListId: The list identifier of the list (1) containing the specified field (1).

@FieldId: The field identifier of the specified field (1).

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

Value	Description
0	Successful execution.
1359	An internal error occurred.
5069	The unique column constraint on the field (1) in the list (1) could not be set because the specified list (1) prevents a user from viewing list items created by another user.

Result sets: MUST NOT return any result sets.

3.1.4.71 **proc_InsertLookupRelationship**

The **proc_InsertLookupRelationship** stored procedure is called to add a relationship lookup field to a list (1), and to optionally specify a relationship delete behavior. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_InsertLookupRelationship (  
    @SiteId      uniqueidentifier,
```

```

@WebId          uniqueidentifier,
@ListId         uniqueidentifier,
@FieldId       uniqueidentifier,
@LookupListId  uniqueidentifier,
@DeleteBehavior tinyint,
@RequestGuid    uniqueidentifier = null OUTPUT
);

```

@SiteId: The site collection identifier of the site collection which contains the specified list (1).

@WebId: The site identifier of the site (2) which contains the specified list (1).

@ListId: The list identifier of the list (1) containing the specified relationship lookup field.

@FieldId: The field identifier of the specified relationship lookup field.

@LookupListId: The list identifier of the target list (1) for the specified relationship lookup field.

@DeleteBehavior: A value which specifies the type of relationship delete behavior that **MUST** be set on the specified relationship lookup field. No relationship delete behavior **MUST** be set if the value is zero. If the value is 1 or 2, the corresponding relationship delete behavior specified in [Relationship Delete Behavior Type](#) **MUST** be set.

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

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

Value	Description
0	Successful execution.
19	The target list (1) specified by @LookupListId does not exist and @DeleteBehavior is 1 or 2.
155	The list server template (as specified in [MS-WSSFO2] section 2.2.3.12) of the list specified by @ListId or the target list (1) specified by @LookupListId is 202 and @DeleteBehavior is either 1 or 2.
1359	An internal error occurred.

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

3.1.4.72 **proc_IsFieldALookupRelationship**

The **proc_IsFieldALookupRelationship** stored procedure is called to verify whether a specified lookup field in a list (1) is a relationship lookup field. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_IsFieldALookupRelationship (
@SiteId          uniqueidentifier,
@ListId         uniqueidentifier,
@FieldId       uniqueidentifier,
@RequestGuid    uniqueidentifier = null OUTPUT
);

```

@SiteId: The site collection identifier of the site collection which contains the specified list (1).

@ListId: The list identifier of the list (1) containing the specified lookup field.

@FieldId: The field identifier of the specified lookup field.

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

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

Value	Description
0	The lookup field specified by @FieldId is not a relationship lookup field.
1	The lookup field specified by @FieldId is a relationship lookup field.

Result sets: MUST NOT return any result sets.

3.1.4.73 **proc_IsSiteScheduledForDeletion**

The **proc_IsSiteScheduledForDeletion** stored procedure is called to verify if a specific site collection is scheduled for deletion. The T-SQL syntax for the stored procedure is as follows:

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

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

Return values: returns an integer return code which MUST be listed in the following table.

Value	Description
0	The site collection is not scheduled for deletion.
1	The site collection is scheduled for deletion

Result sets: MUST not return any result sets.

3.1.4.74 **proc_ListThemeFiles**

The **proc_ListThemeFiles** stored procedure is called to retrieve information about all related files for a given theme. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ListThemeFiles(  
    @SiteId          uniqueidentifier,  
    @WebUrl          nvarchar(260),  
    @ThemeName       nvarchar(128),  
    @RequestGuid     uniqueidentifier = NULL OUTPUT  
);
```

@SiteId: The site collection identifier of the site collection which contains the site (2) specified by the **@WebUrl** parameter.

@WebUrl: The Uniform Resource Locator (URL) in store-relative form of the site (2) that contains the theme specified by the @ThemeName stored procedure parameter. Specifying NULL for this

parameter will retrieve the theme files for the top-level site of the site collection specified by the **@SiteId** parameter.

@ThemeName: The Unicode name of the theme to retrieve the files.

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

Return values: An integer that MUST be zero.

Result sets: MUST return one result set.

3.1.4.74.1 Theme Files Information Result Set

The **Theme Files Information Result Set** contains information about the files associated with the theme specified by the parameter *@ThemeName*, for a site (2) specified by the parameter *@WebUrl* that belongs to the site collection specified by the parameter *@SiteId*. There MUST be one row for each such file.

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

```
{FileURL}          nvarchar(385),
Content            varbinary(max),
HasStream          int,
SetupPathVersion  tinyint,
SetupPath          nvarchar(255);
```

{FileURL}: The URL of a file in store-relative form for the theme.

Content: For uncustomized files associated with the theme, this MUST be NULL. For files that are not uncustomized, this MUST be the content of the file associated with the theme as a binary image.

HasStream: MUST be "1" if the theme file is uncustomized. Otherwise, MUST be zero ("0") if the theme file is not uncustomized.

SetupPathVersion: This MUST be one of the values in the following table with regard to the file associated with the theme.

Value	Description
NULL	This file has never been uncustomized.
"2"	The path of the file is relative to the installation location of Windows® SharePoint® Services 2.0 on the front-end Web server. For example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\60.
"3"	The path of the file is relative to the installation location of Windows® SharePoint® Services 3.0 on the front-end Web server. For example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\12.
"4"	The path of the file is relative to the installation location of Microsoft® SharePoint® Foundation 2010 on the front-end Web server. For example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\14.

SetupPath: For a file associated with the theme that has never been uncustomized, this MUST be the path fragment relative to the path returned by the **SetupPathVersion** column value. Taken together, the **SetupPathVersion** and **SetupPath** columns determine where these files are located on the front-end Web server's file system. For example, if **SetupPathVersion** is "4" and

SetupPath is "themes\Lichen\LICHEN.INF", the path is Program Files\Common Files\Microsoft Shared\Web Server Extensions\14\template\themes\Lichen\LICHEN.INF. This MUST be NULL if the file has never been customized.

3.1.4.75 proc_ListThemes

The **proc_ListThemes** stored procedure is called to return information about a theme that is contained within a site (2). The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ListThemes(  
    @SiteId            uniqueidentifier,  
    @WebUrl            nvarchar(260)  
    @RequestGuid      uniqueidentifier = NULL OUTPUT  
);
```

@SiteId: The site collection identifier of the site collection which contains the site (2) specified by the **@WebUrl** parameter.

@WebUrl: The URL in store-relative form of the site (2) whose theme information will be retrieved. Specifying NULL for this parameter will retrieve the theme information for the top-level site of the site collection specified by the **@SiteId** parameter.

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

Return values: An integer that MUST be zero.

Result sets: MUST return one result set.

3.1.4.75.1 Theme Information Result Set

The **Theme Information Result Set** contains information about the theme contained within the site (2) specified by the parameter **@WebUrl** that belongs to the site collection specified by the parameter **@SiteId**. For combinations of the **@WebUrl** and **@SiteId** parameters that define an existing site (2), one row MUST be returned. Otherwise, an empty result set MUST be returned.

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

```
LeafName            nvarchar(128),  
SetupPathVersion    tinyint,  
SetupPath           nvarchar(255),  
SetupPathUser       nvarchar(255),  
Content             varbinary(max);
```

LeafName: The theme name.

SetupPathVersion: This MUST be one of the values in the following table.

Value	Description
"2"	The path to the .inf file that describes the theme that describes the theme returned by the Theme Information Result Set SetupPath column is relative to the installation location of Windows® SharePoint® Services 2.0 on the front-end Web server. For example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\60.
"3"	The path to the .inf file that describes the theme returned by the Theme Information Result

Value	Description
	Set SetupPath column is relative to the installation location of Windows® SharePoint® Services 3.0 on the front-end Web server. For example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\12.
"4"	The path to the .inf file that describes the theme returned by the Theme Information Result Set SetupPath column is relative to the installation location of Microsoft® SharePoint® Foundation 2010 on the front-end Web server. For example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\14.

SetupPath: This MUST be the path fragment of the .inf file that describes the theme that describes the theme. Taken together, the **SetupPathVersion** and **SetupPath** columns determine where the .inf file that describes the theme is located on the front-end Web server's file system. For example, if **SetupPathVersion** is "4" and **SetupPath** is "themes\Lichen\LICHEN.INF", the path to the Theme INF File is Program Files\Common Files\Microsoft Shared\Web Server Extensions\14\template\themes\Lichen\LICHEN.INF.

SetupPathUser: This contains the login name of the user that created the theme.

Content: If the .inf file that describes the theme is uncustomized, this MUST be NULL. If the .inf file that describes the theme is not uncustomized, this MUST be the content of the .inf file that describes the theme.

3.1.4.76 proc_LoadTheme

The **proc_LoadTheme** stored procedure is called to return data about certain files related to a given theme. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_LoadTheme (
    @WebSiteId          uniqueidentifier,
    @ThemesDir          nvarchar(256),
    @ThemeName          nvarchar(128),
    @GraphicCSS         nvarchar(128),
    @ColorCSS           nvarchar(128),
    @ExtCSS             nvarchar(128),
    @NeedThemesInf     bit,
    @RequestGuid        uniqueidentifier = NULL OUTPUT
);

```

@WebSiteId: The site collection identifier of the site collection from which to retrieve the theme data.

@ThemesDir: The URL in store-relative form of the theme. This stored procedure parameter MUST be one of the values in the following table.

Value	Description
_themes	Retrieves the theme data for the top-level site of the site collection specified by the @WebSiteId stored procedure parameter.
[Site]/_themes	Retrieves the theme data for a site (2) that belongs to the site collection specified by the @WebSiteId stored procedure parameter. For example, if "Contoso" is the name of a site (2) that belongs to the site collection, then to retrieve the theme data for the "Contoso" site (2), the @ThemesDir parameter would be: 'Contoso/_themes'

@ThemeName: Unicode name of the theme.

@GraphicCSS: The Unicode file name of a cascading style sheet (CSS) that defines how text is displayed in a user interface for a theme in the navigation bars. For example, this CSS might define the images used for displaying hyperlinks, the font-family, font-size, font-style, font-weight, font color, text-align of text to be displayed in the navigation bars. The value **MUST** be one of the values in the following table.

Value	Description
graph0.css	The CSS defined for theme text that is not active in a link bar . For example, a link in is not active if the end user has not interacted with it by clicking on the link with a mouse.
graph1.css	The CSS defined for theme text that is active in a link bar. For example, a link in a theme is said to be active when the end user clicks on the link with a mouse.

@ColorCSS: The Unicode file name of a CSS that defines the color for text, hyperlinks, and background images in a user interface for a theme. The value **MUST** be one of the values in the following table.

Value	Description
color0.css	The CSS for the normal color set for the theme. The normal color set is comprised of colors that are traditionally used and more aptly supported.
color1.css	The CSS for the vivid color set for the theme. The vivid color set expands the normal color set with a brighter set of colors.

@ExtCSS: The masked Unicode file name that matches a set of CSS files whose location or contents will be returned. Here, "masked" refers to the fact that @ExtCSS **MUST** be specified exactly using the following pattern: %extension.css.

This stored procedure will interpret this and return any CSS whose file name suffix is "extension.css" and where the percent sign (%) can be replaced with any character or characters. For example, the CSS file name of 'contosoextension.css' would be returned. Those CSS definitions that cannot be defined in the CSS files provided by the **@ColorCSS** and **@GraphicCSS** stored procedure parameters **MUST** be defined in the CSS files that match this pattern.

@NeedThemesInf: Specifies if this stored procedure has to return information about the .inf file that describes the theme. The value **MUST** be one of the values in the following table.

Value	Description
0	MUST NOT return information about the .inf file that describes the theme.
1	MUST return information about the .inf file that describes the theme.

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

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

Value	Description
0	The stored procedure parameter @NeedThemesInf has value zero or the stored procedure parameter @NeedThemesInf has value 1 and the corresponding .inf file that describes the theme was found.

Value	Description
1	The stored procedure parameter @NeedThemesInf has value 1 which indicated that the .inf file that describes the theme was specifically requested, but it was not found.

Result Sets: MUST return two result sets as described in the following sections.

3.1.4.76.1 Theme Files Information Result Set

The **Theme Files Information Result Set** contains information about the following theme files:

1. theme.css
2. custom.css
3. The Unicode given by the parameter **@GraphicCSS**.
4. The Unicode given by the parameter **@ColorCSS**.
5. The Unicode that matches the pattern given by the parameter **@ExtCSS**.
6. The Unicode given by concatenation of the value from the parameter **@ThemeName** and the file extension .utf8.
7. The Unicode given by concatenation of the value from the parameter **@ThemeName** and extension .inf if the file with the extension .utf8 is not found.

Information about files that are found according to the preceding criteria will be returned as a row per qualifying file in the result set. Note that more than one row can be returned for cascading style sheet (CSS) file names that match the criteria specified by the **@ExtCSS** parameter. The T-SQL syntax for the result set is as follows:

```
LeafName          nvarchar(128),
SetupPathVersion  tinyint,
SetupPath         nvarchar(255),
SetupPathUser     nvarchar(255),
Content           varbinary(max),
UsesExternalStorage bit;
```

LeafName: The theme name.

SetupPathVersion: This MUST be one of the values in the following table.

Value	Description
2	The path to the theme file is relative to the installation location of Windows® SharePoint® Services 2.0 on the front-end Web server. For example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\60.
3	The path to the theme file is relative to the installation location of Windows® SharePoint® Services 3.0 on the front-end Web server. For example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\12.
4	The path to the theme file is relative to the installation location of Microsoft® SharePoint® Foundation 2010 on the front-end Web server. For example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\14.

SetupPath: This MUST be the path fragment of the theme file. Taken together, the SetupPathVersion and SetupPath columns determine where the theme file is located on the front-end Web server's file system. For example, if the SetupPathVersion is 4 and the SetupPath is themes\Lichen\LICHEN.INF, then the path would be Program Files\Common Files\Microsoft Shared\Web Server Extensions\14\template\themes\Lichen\LICHEN.INF.

SetupPathUser: This contains the login name of the user that created the theme.

Content: If the theme file is uncustomized, this MUST be NULL. If the theme file is not uncustomized, this MUST be the content of the theme file.

UsesExternalStorage: This MUST be "1" if the document content is contained in an external storage or the document has a remote BLOB storage identifier; else MUST be zero.

3.1.4.76.2 Theme INF File Information Result Set

The **Theme INF File Information Result Set** contains information about the .inf file that describes the theme. If the .inf file that describes the theme can be found, then one row MUST be returned. If the .inf file that describes the theme cannot be found, then no rows MUST be returned. The name of the .inf file that describes the theme MUST be the Unicode obtained by concatenating the value from the parameter **@ThemeName** and the extension '.inf'. The T-SQL syntax for the result set is as follows:

```
LeafName          nvarchar(128),
SetupPathVersion  tinyint,
SetupPath         nvarchar(255),
SetupPathUser     nvarchar(255),
Content           varbinary(max),
UsesExternalStorage bit;
```

The semantic of these parameters is the same as for those in the previously defined **Theme Files Information Result Set**.

3.1.4.77 proc_LogChange

The **proc_LogChange** stored procedure is called to store the information about an event (1) that is either triggered by the user or triggered by the system.

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

```
PROCEDURE proc_LogChange (
    @SiteId          uniqueidentifier,
    @WebId           uniqueidentifier,
    @ListId          uniqueidentifier,
    @ItemId          int,
    @DocId           uniqueidentifier,
    @Guid0           uniqueidentifier,
    @Int0            int,
    @FullUrl         nvarchar(260),
    @EventType       int,
    @ObjectType      int,
    @TimeLastModifiedIncoming datetime,
    @ItemName        nvarchar(255) = NULL,
    @Int1            int = NULL,
    @DocClientId     varbinary(16) = NULL,
    @RequestGuid     uniqueidentifier = NULL OUTPUT
```

);

- @SiteId:** This value is a change log **SiteId** (section [2.2.2.11](#)).
- @WebId:** This value is a change log **WebId** (section [2.2.2.12](#)).
- @ListId:** This value is a change log **ListId** (section [2.2.2.1](#)).
- @ItemId:** This value is a change log **ItemId** (section [2.2.2.2](#)).
- @DocId:** This value is a change log **DocId** (section [2.2.2.3](#)).
- @Guid0:** This value is a change log **Guid0** (section [2.2.2.4](#)).
- @Int0:** This value is a change log **Int0** (section [2.2.2.5](#)).
- @FullUrl:** This value is a change log **ItemFullUrl** (section [2.2.2.7](#)).
- @EventType:** An integer that represents the [Event Type Flags](#).
- @ObjectType:** An integer that represents the [Event Object Type Flags](#).
- @TimeLastModifiedIncoming:** This value is change log **TimeLastModified** (section [2.2.2.8](#)).
- @ItemName:** This value is a change log **ItemName** (section [2.2.2.9](#))
- @Int1:** This value is a change log **Int1** (section [2.2.2.10](#))
- @DocClientId:** An optional GUID which specifies the identity of this object for purposes of client synchronization.
- @RequestGuid:** The optional request identifier for the current request.
- Return Code Values:** An integer that MUST be zero.
- Result Sets:** MUST NOT return any result sets.

3.1.4.78 **proc_LogChangeForFileFragments**

The **proc_LogChangeForFileFragments** stored procedure is called to store the information about an event (1) that was triggered by the creation of a file fragment.

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

```
PROCEDURE proc_LogChangeForFileFragments (  
    @SiteId                uniqueidentifier,  
    @WebId                 uniqueidentifier,  
    @DocId                 uniqueidentifier,  
    @Partition             tinyint  
);
```

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

@WebId: The site identifier of the site (2) in which contains the specified file fragment.

@DocId: The document identifier (2) of the document associated with the file fragment.

@Partition: The identifier for a file fragment partition of the partition (2) to which the file fragment belongs.

Return Code Values: An integer that MUST be zero.

Result Sets: MUST NOT return any result sets.

3.1.4.79 **proc_PatchLinkForFile**

The **proc_PatchLinkForFile** stored procedure is called to update the directory name and leaf name of a requested link for a file, to prepare for delayed link fixup. It calls **proc_DirtyDependents** which marks all dependent documents and links as dirty. See [\[MS-WSSFO2\]](#) section 3.1.5.15 for **proc_DirtyDependents**. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_PatchLinkForFile(  
    @SiteId          uniqueidentifier,  
    @WebId           uniqueidentifier,  
    @DirName         nvarchar(256),  
    @LeafName        nvarchar(128),  
    @OldLinkDirName  nvarchar(256),  
    @OldLinkLeafName nvarchar(128),  
    @OldServerRel    bit,  
    @NewLinkDirName  nvarchar(256),  
    @NewLinkLeafName nvarchar(128),  
    @NewServerRel    bit,  
    @PatchPrefix     bit,  
    @DocUpdateFlags  int,  
    @RequestGuid     uniqueidentifier = NULL OUTPUT  
);
```

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

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

@DirName: The directory containing the file. MUST NOT be NULL.

@LeafName: The file name. MUST NOT be NULL.

@OldLinkDirName: The directory containing the linked file, prior to running the stored procedure. MUST NOT be NULL.

@OldLinkLeafName: The linked file's name, prior to running the stored procedure. MUST NOT be NULL.

@OldServerRel: Indicates if old link is server-relative URL. If this is 1, **@NewServerRel** and **@PatchPrefix** MUST also be 1.

@NewLinkDirName: Target directory name for the linked file. MUST NOT be NULL.

@NewLinkLeafName: Target file name for the linked file. MUST NOT be NULL.

@NewServerRel: Indicates if new link is server-relative URL. If this is 1, **@OldServerRel** and **@PatchPrefix** MUST also be 1.

@PatchPrefix: If this is 1, the whole URL of the directory is updated. If this is 1, **@OldServerRel** and **@NewServerRel** MUST also be 1.

@DocUpdateFlags: An integer representing the Put Flags Type as defined in [\[MS-WSSFO2\]](#) section 2.2.2.7 that MUST be used for this update.

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

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

Value	Description
0	Successful execution.
3	Cannot find the requested document in the site collection, or the site collection does not exist.
1150	Error updating link data.

Result sets: MUST NOT return any result sets.

3.1.4.80 **proc_PatchLinkForWeb**

The **proc_PatchLinkForWeb** stored procedure is called to patch links within a site (2), preparing it for delayed link fixup. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_PatchLinkForWeb(
    @SiteId                uniqueidentifier,
    @WebId                 uniqueidentifier,
    @OldLinkDirName        nvarchar(256),
    @OldLinkLeafName       nvarchar(128),
    @NewLinkDirName        nvarchar(256),
    @NewLinkLeafName       nvarchar(128),
    @PatchFlags            int = 0,
    @RequestGuid           uniqueidentifier = NULL OUTPUT
);
```

@SiteId: The site collection identifier of the site collection which contains the specified site (2).

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

@OldLinkDirName: The directory name containing the linked site (2). MUST NOT be NULL.

@OldLinkLeafName: The leaf name containing the linked site (2). MUST NOT be NULL.

@NewLinkDirName: The new directory name for the linked site (2). MUST NOT be NULL.

@NewLinkLeafName: The new leaf name for the linked site (2). MUST NOT be NULL.

@PatchFlags: A 32-bit mask containing control flags. This can have zero, one, or two flags set. This parameter is optional with a default value of zero. The valid flags are described in the following table.

Value	Meaning
0x00000000	Default value

Value	Meaning
0x00000001	Indicates directory names are more than one level above the leaf. The new link directory name replaces part of the old link directory name, and lower-level sites (2) MUST be updated. For example, the domain name is changed but the structure below remains the same.
0x00000002	Indicates that the modification date of the site (2) MUST be updated.

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

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

Value	Description
0	Successful execution.
3	Cannot find the requested site (2) in the site collection, or the site collection does not exist.
1150	Error updating link data.

Result sets: MUST NOT return any result sets.

3.1.4.81 **proc_ReadSharedAccessRequests**

The **proc_ReadSharedAccessRequests** stored procedure returns the list of document identifiers (2) for which co-authoring transition requests has been made. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ReadSharedAccessRequests (
    @Expired          datetime
);
```

@Expired: The time when this stored procedure considers the co-authoring transition request to have expired. Any request that was made before this time is not returned in the result set.

Return Code Values: An integer that MUST be zero.

Result Sets: The stored procedure must return a result set of zero or more rows of the **Shared Access Requests Result Set** (section [3.1.4.81.1](#)).

3.1.4.81.1 **Shared Access Requests Result Set**

The **Shared Access Requests Result Set** contains the list of document identifiers (2) for which co-authoring transition requests have been made. The T-SQL syntax for the result set is as follows:

```
DocumentId uniqueidentifier NOT NULL
```

DocumentId: The document identifier (2) of the document for which a co-authoring transition request has been made.

3.1.4.82 proc_RefreshCheckout

The **proc_RefreshCheckout** stored procedure is called to renew the short-term check-out on the specified document for the specified user. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_RefreshCheckout (
    @SiteId                uniqueidentifier,
    @DirName                nvarchar(256),
    @LeafName              nvarchar(128),
    @SystemId              varbinary(512),
    @CheckoutTimeout       int,
    @RequestGuid           uniqueidentifier = NULL OUTPUT
);
```

@SiteId: The site collection identifier of the site collection which contains the specified checked out document.

@DirName: The directory name of the specified checked out document.

@LeafName: The leaf name of the specified checked out document.

@SystemId: The SystemID of the user who has checked out the specified checked out document.

@CheckoutTimeout: New timeout in minutes for short-term check-out of the specified checked out document. It MUST NOT be NULL.

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

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

Value	Description
0	Success. The check-out on the specified checked out document was successfully refreshed.
2	File not found. A checked out document corresponding to the specified @SiteId, @DirName, @LeafName, and @SystemId parameters does not exist or it exists but is not checked out by the user specified by @SystemId.
5	There is no user corresponding to @SystemId.

Result sets: MUST return the result sets described in the following sections under the specified conditions,

3.1.4.82.1 Document Metadata Result Set

The **Document Metadata Result Set** MUST be returned only if the document specified by **@SiteId**, **@DirName** and **@LeafName** exists for the user specified by **@SystemId**. See [\[MS-WSSFO2\]](#) section 2.2.5.6. All values in the column named **CacheParseId** MUST be NULL.

3.1.4.82.2 NULL Result Set

The **NULL Result Set** MUST be returned only if the document specified by **@SiteId**, **@DirName** and **@LeafName** exists for the user specified by **@SystemId**. See the **NULL Result Set** definition in [\[MS-WSSFO2\]](#) section 3.1.5.17.3.

3.1.4.83 proc_RemoveJunctions

The **proc_RemoveJunctions** stored procedure is called to remove a value from the set of values of a multivalued lookup field of a specified list item in a list (1). The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_RemoveJunctions(  
    @SiteId                uniqueidentifier,  
    @DirName                nvarchar(256),  
    @LeafName              nvarchar(128),  
    @FieldId               uniqueidentifier,  
    @DeleteTransactionId   varbinary(16) = 0x,  
    @Level                  tinyint = 1,  
    @IsCurrentVersion       bit = 1,  
    @CalculatedVersion     int = 0  
    @RequestGuid           uniqueidentifier = NULL OUTPUT  
);
```

@SiteId: The site collection identifier of the site collection which contains the list (1) that contains the multivalued lookup field specified by **@FieldId**.

@DirName: The directory name of the specified list item.

@LeafName: The leaf name of the specified list item.

@FieldId: The field identifier of the specified multivalued lookup field.

@DeleteTransactionId: This parameter MUST be 0x.

@Level: The publishing level of this list item.

@IsCurrentVersion: A bit flag specifying whether the specified row belongs to the current version of the list item.

@CalculatedVersion: This parameter MUST be zero.

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

Return Values: This store procedure MUST return zero upon completion.

Result Sets: MUST NOT return any result sets.

3.1.4.84 proc_RemoveSharedAccessRequest

The **proc_RemoveSharedAccessRequest** stored procedure is called to remove the co-authoring transition request for a document. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_RemoveSharedAccessRequest (  
    @SiteId                uniqueidentifier,  
    @DocDirName            nvarchar(256),  
    @DocLeafName           nvarchar(128),  
    @DocId                 uniqueidentifier OUTPUT,  
    @RequestGuid           uniqueidentifier = NULL OUTPUT  
);
```


@SiteId: The site collection identifier of the site collection which contains the document for which the co-authoring transition request should be removed.

@DocDirName: The directory name of the document.

@DocLeafName: The leaf name of the document.

@DocId: An output parameter containing the document identifier (2) of the document for which the co-authoring transition request should be removed.

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

Return values: An integer that MUST be zero upon successful completion. The stored procedure MUST return a nonzero return value if an error occurred.

Result sets: MUST NOT return any result sets.

3.1.4.85 `proc_RenameHostHeaderSite`

The **`proc_RenameHostHeaderSite`** stored procedure is called to change the URL of a host-named site collection to a new URL. It is executed in the **configuration database** and the content database. The stored procedure also marks all documents as dirty in this site (2) which have forward links with server-relative URL. The procedure assumes that no site (2) already exists at that path.

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

```
PROCEDURE proc_RenameHostHeaderSite(  
    @SiteId                uniqueidentifier,  
    @HostHeader            nvarchar(128)  
);
```

@SiteId: The site collection identifier of the site collection. MUST NOT be NULL.

@HostHeader: The **Internet Information Services (IIS)** host header for the Internet Information Services (IIS) **Web application (1)** containing the site collection. The host header is an optional property of an Internet Information Services (IIS) Web application (1). MUST be NULL, or a host header string.

Return values: An integer that the protocol client MUST ignore.

Result sets: MUST NOT return any result sets.

3.1.4.86 `proc_RenameSite`

The **`proc_RenameSite`** stored procedure is called to move a site collection to a different URL. All permutations of moves in root and non-root folder structure are supported; that is, 'sites/one' to 'sites/onenew', or 'sites/one' to 'sites/another/one', and so forth. The stored procedure assumes that no site collection already exists at that path. It does not update the site collection URL entry in the configuration database. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_RenameSite(  
    @SiteId                uniqueidentifier,  
    @OldUrl                nvarchar(260),  
    @NewUrl                nvarchar(260),  
    @SiteFullUrl           nvarchar(260),
```

```

    @useHostHeaderSite    bit,
    @RequestGuid          uniqueidentifier = NULL OUTPUT
);

```

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

@OldUrl: The URL of the site collection before the stored procedure is called. It can be the URL of the top-level site (2) or any level underneath that.

@NewUrl: The URL of the site collection after it has been renamed. The new URL MUST be in the same Web application (1).

@SiteFullUrl: The response URL.

@useHostHeaderSite: Indicates if the site collection uses a host header. If zero, the server MUST update the site collection to not use a host header. Otherwise, this flag MUST be ignored.

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

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

Value	Description
0	Successful execution.
206	The new URL of one or more of the objects of the site (2) would be longer than 260 characters.
1003	A site (2) with the identifier @SiteId was not found, or the operation caused a database error.
1150	A database error occurred when marking lists (1) as dirty for lists (1) which have list items with forward links to any documents whose URL will be changed as a result of this operation.

Result sets: MUST NOT return any result sets.

3.1.4.87 proc_SaveFileFormatMetaInfo

The **proc_SaveFileFormatMetaInfo** stored procedure is called to update the file format metadata for an existing document. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_SaveFileFormatMetaInfo (
    @SiteId          uniqueidentifier,
    @DocId           uniqueidentifier,
    @Level           tinyint,
    @DocVersion      int,
    @ContentVersion  int,
    @FFM             varbinary(max),
    @FFMSize         int,
    @FailIfExists    bit
);

```

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

@DocId: The document identifier (2) of the document to update.

@Level: The publishing level of the document to update.

@DocVersion: The current internal version number of the document. The file format metadata for the document MUST not be updated if **@DocVersion** is not NULL and the current version number of the document is not NULL and **@DocVersion** is not equal to the current version number.

@ContentVersion: The content version of the document to update. The file format metadata for the document MUST not be updated if **@ContentVersion** is not NULL and does not equal the current content version number of the document.

@FFM: The contents of the file format metadata.

@FFMSize: The size, in bytes, of the file format metadata. This value SHOULD be the size of **@FFM**, in bytes.

@FailIfExists: Specifies whether file format metadata can exist for the document currently. If this value is 1, there MUST not be file format metadata for the document that is not NULL.

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

Value	Description
0	Successful execution.
2	The document was not found.
21	The document was not updated.
30	An IO error or constraint violation occurred when finding the existing document.
183	@FailIfExists is 1 and the existing document has file format metadata that is not NULL.

Result sets: MUST NOT return any result sets.

3.1.4.88 **proc_SaveFileFragmentById**

The **proc_SaveFileFragmentById** stored procedure is called to insert or update a file fragment for an existing document. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_SaveFileFragmentById (  
    @SiteId          uniqueidentifier,  
    @DocId           uniqueidentifier,  
    @DocVersion      int,  
    @UserId          int,  
    @Id              bigint,  
    @Partition       tinyint,  
    @Tag             varbinary(40),  
    @BlobData        varbinary(max),  
    @BlobSize        int,  
    @QuotaChange     int OUTPUT,  
    @NewId           bigint OUTPUT  
);
```

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

@DocId: The document identifier (2) of the document to be associated with the file fragments being updated.

@DocVersion: This parameter MUST be ignored.

@UserId: This parameter MUST be ignored.

@Id: The file fragment identifier of the file fragment to insert/update. If this parameter is NULL, a new file fragment MUST be inserted. Otherwise an existing file fragment MUST be updated.

@Partition: The identifier for a file fragment partition of the file fragment partition to which the inserted or updated file fragment belongs.

@Tag: The file fragment tag of the file fragment to insert or update.

@BlobData: The data of the file fragment to be inserted or updated.

@BlobSize: The size in bytes of the file fragment data to be inserted or updated. This value SHOULD be the size of **@BlobData**, in bytes. If **@BlobSize** is not equal to the size in bytes of **@BlobData**, all of the bytes of **@BlobData** MUST still be written.

@QuotaChange: The net amount of change to the file fragment data, in bytes. If a new file fragment is being inserted, this parameter MUST be set to **@BlobSize**. Otherwise, this parameter MUST be set to the difference of the current data size and **@BlobSize**, in bytes. This parameter MUST be set to zero if the file fragment was not inserted or updated.

@NewId: If a new file fragment is being inserted, this value MUST be set to the file fragment identifier assigned to it. Otherwise this value MUST be set to **@Id**.

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

Value	Description
0	Successful execution.
33	The file fragment being updated could not be found.
4317	The file fragment being inserted or updated resulted in an IO error or constraint violation.

Result sets: MUST NOT return any result sets.

3.1.4.89 **proc_SaveFileFragmentByTag**

The **proc_SaveFileFragmentByTag** stored procedure is called to insert a file fragment for a document based on a consistency check of an existing of file fragment tag value. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_SaveFileFragmentByTag (  
    @SiteId          uniqueidentifier,  
    @DocId           uniqueidentifier,  
    @DocVersion     int,  
    @UserId         int,  
    @OldId          bigint,  
    @Partition      tinyint,  
    @Tag            varbinary(40),  
    @BlobData       varbinary(max),  
    @BlobSize       int,  
    @QuotaChange    int OUTPUT,  
    @NewId          bigint OUTPUT  
);
```

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

@DocId: The document identifier (2) of the document to be associated with the file fragments being inserted.

@DocVersion: This parameter MUST be ignored.

@UserId: This parameter MUST be ignored.

@OldId: The file fragment identifier used to do a consistency check. If this value is NULL, the file fragment MUST only be inserted if no current file fragment has values for document identifier (2), Identifier for a file fragment partition, and file fragment tag corresponding to **@DocId**, **@Partition**, **@Tag** values. If this value is not NULL, the file fragment MUST only be inserted if a current file fragment exists with values **@DocId**, **@Partition**, **@Tag** and maximum value of file fragment identifier is **@OldId**.

@Partition: The Identifier for a file fragment partition of the file fragment partition to which the inserted file fragment belongs.

@Tag: The file fragment tag of the file fragment to insert, also used for consistency check with **@OldId**.

@BlobData: The data of the file fragment to be inserted.

@BlobSize: The size in bytes of the file fragment data to be inserted/updated. This value SHOULD be the size of **@BlobData**, in bytes. If it is not, the value MUST still be updated as the current size of the data of the file fragment.

@QuotaChange: The net amount of change to the file fragment data, in bytes. This parameter MUST be set to **@BlobSize** if the file fragment was inserted. If the file fragment was not inserted, this parameter MUST be set to NULL.

@NewId: This value MUST be set to the file fragment identifier assigned to the newly inserted file fragment, else it MUST be NULL.

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

Value	Description
0	Successful execution.
33	If @OldId is NOT NULL, a file fragment with values @DocId, @Partition, @Tag and maximum file fragment identifier value @OldId could not be found. If @OldId is NULL, the file fragment with values @DocId, @Partition, and @Tag was found.
4317	The file fragment being inserted resulted in a IO error or constraint violation.

Result sets: MUST NOT return any result sets.

3.1.4.90 **proc_SetListDataSource**

The **proc_SetListDataSource** stored procedure is called to set the data source and Entity identifier for a list (1). The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_SetListDataSource (  
    @ListId          uniqueidentifier,  
    @DataSource      nvarchar(max),
```

```

    @EntityId          int,
    @RequestGuid       uniqueidentifier = NULL OUTPUT
);

```

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

@DataSource: A string that MUST contain the XML fragment for the data source of the list (1). The schema of this fragment is defined by [List Data Source](#).

@EntityId: The identifier for the Entity of the list (1).

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

Return values: An integer that MUST be zero.

Result sets: MUST NOT return any result sets.

3.1.4.91 `proc_SetNextId`

The **proc_SetNextId** stored procedure is called to update the next available identifier of an existing list (1). The next available identifier is an integer identifier for the next new list item in the list (1). The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_SetNextId(
    @WebId              uniqueidentifier,
    @ListId             uniqueidentifier,
    @NextAvailableId   int,
    @RequestGuid       uniqueidentifier = NULL OUTPUT
);

```

@WebId: The site identifier of the site (2) which contains the specified list (1).

@ListId: The list identifier of the list (1) to set the next available identifier.

@NextAvailableId: The integer identifier for the next new list item in the list (1). If the **@NextAvailableId** is greater than the current value of the next available identifier for the requested list (1), the **proc_SetNextId** MUST update the next available identifier of the requested list (1) to be **@NextAvailableId**. Otherwise, the **proc_SetNextId** MUST not update the next available identifier of the requested list (1).

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

Return values: An integer that MUST be zero.

Result sets: MUST NOT return any result sets.

3.1.4.92 `proc_StartUndirtyList`

The **proc_StartUndirtyList** stored procedure is called to start a new link fixup operation. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_StartUndirtyList(
    @WebId              uniqueidentifier,
    @ListId             uniqueidentifier,
    @CacheParseId       uniqueidentifier,

```

```

    @RequestGuid      uniqueidentifier = NULL OUTPUT
);

```

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

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

@CacheParseId: GUID representing new link fixup operation. This is used to assure that no one else has updated or started to update the document during the link fixup operation. MUST NOT be NULL.

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

Return values: An integer that MUST be zero.

Result sets: MUST return the result set described in the following section.

3.1.4.92.1 Cache Parse Identifier Result Set

The **Cache Parse Identifier Result Set** contains the effective cache parse identifier for the new link fixup operation. The T-SQL syntax for the result set is as follows:

```

tp_CacheParseId uniqueidentifier NOT NULL

```

tp_CacheParseId: GUID representing the new link fixup operation. This is used to assure that no one else has updated or started to update the document during the link fixup operation.

3.1.4.93 proc_TakeOfflineDocument

The **proc_TakeOfflineDocument** stored procedure is called to take the last published version or major version of the document offline. The document will then be visible only to users with permission to edit the document, and no longer visible to users browsing the Site. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_TakeOfflineDocument (
    @SiteId          uniqueidentifier,
    @WebId           uniqueidentifier,
    @DirName         nvarchar(256),
    @LeafName        nvarchar(128),
    @CreateVersion   bit,
    @EnableMinorVersions bit,
    @Moderated       bit,
    @UserId          int,
    @MaxMajorVersion int,
    @MaxMajorMinorVersion int,
    @RequestGuid     uniqueidentifier = NULL OUTPUT
);

```

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

@WebId: The site identifier of the site (2) which contains the specified document.

@DirName: The directory name containing the specified document.

@LeafName: The leaf name of the specified document.

@CreateVersion: A bit flag specifying whether the document library containing the document has version numbering enabled. If version numbering is enabled for the document library, this parameter MUST be 1; otherwise, this parameter MUST be zero. If the document is not in a document library, this parameter MUST be zero. This parameter MUST NOT be NULL.

@EnableMinorVersions: A bit flag specifying whether the document library containing the document has minor version numbering enabled. If minor version numbering is enabled for the document library containing the document, this parameter MUST be set to 1; otherwise, this parameter MUST be set to zero. If the document is not in a document library, this parameter MUST be set to zero. This parameter MUST NOT be NULL.

@Moderated: A bit flag specifying whether the document library containing the document is a moderated object. If document library containing the document is a moderated object, this parameter MUST be set to 1. In all other cases, it MUST be zero. This parameter MUST NOT be NULL.

@UserId: The identifier for the current user who is requesting this operation. This value MUST refer to an existing user identifier for the specified site collection.

@MaxMajorVersion: The maximum number of major versions that the document library will track for any document. This is a setting of the document library containing the specified document.

@MaxMajorMinorVersion: The maximum number of major versions that can have associated minor versions of a document in the document library. This is a setting of the document library containing the specified document.

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

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

Value	Description
0	Successful execution.
3	File Not Found. A document having publishing level of published corresponding to the specified @SiteId, @WebId, @DirName and @LeafName was not found; or @EnableMinorVersions is "0" and @Moderated is "0".

Result sets: MUST return the result sets described in the following sections under the specified conditions.

3.1.4.93.1 Document Metadata Result Set

The **Document Metadata Result Set** MUST be returned if the execution is successful. See [\[MS-WSSFO2\]](#), section [2.2.5.6](#). All values in the column named **{CacheParseId}** MUST be NULL.

3.1.4.93.2 Event Receivers Result Set

The **Event Receivers Result Set** MUST be returned if the current version of the specified document is published and the **@WebId** parameter specifies the site collection. This result set MUST contain one row for each event receiver registered for the document with an Event Host Type (see [\[MS-WSSFO2\]](#), section [2.2.3.5](#)).

3.1.4.93.3 NULL Result Set

The **NULL Result Set** MUST be returned if the current version of the specified document is published and the **@WebId** parameter is NULL. See the **NULL Result Set** definition in [\[MS-WSSFO2\]](#) section 3.1.5.17.3.

3.1.4.93.4 Link Info Single Doc Result Set

The **Link Info Single Doc Result Set** MUST be returned if the execution is successful. See the **Link Info Single Doc Result Set** in [\[MS-WSSFO2\]](#) section 3.1.5.7.1.

3.1.4.94 proc_UndirtyListItem

The **proc_UndirtyListItem** stored procedure is called to update the **metadict** of the document and clear the dirty status of the document. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_UndirtyListItem(
    @SiteId          uniqueidentifier,
    @DirName         nvarchar(256),
    @LeafName        nvarchar(128),
    @Level           tinyint,
    @CacheParseId    uniqueidentifier,
    @MetaInfo        varbinary(max),
    @RequestGuid     uniqueidentifier = NULL OUTPUT
);
```

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

@DirName: The directory name of the specified document.

@LeafName: The leaf name of the requested document.

@Level: Indicates the publishing level of the list item.

@CacheParseId: GUID representing a new link fixup operation. This is used to assure that no one else has updated or started to update the document during the link fixup operation. MUST NOT be NULL.

@MetaInfo: A metadict for the document as specified in [\[MS-FPSE\]](#) section 2.2.4.

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

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

Value	Description
0	Successful execution.
3	Specified list (1) or site (2) is not found.
1150	Specified cache parse identifier is not correct; another cache parse identifier is already in use.

Result sets: MUST NOT return any result sets.

3.1.4.95 proc_UpdateEntityIdForList

The **proc_UpdateEntityIdForList** stored procedure is called to change the Entity identifier of the list (1). The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_UpdateEntityIdForList (
    @ListId          uniqueidentifier,
    @EntityId        int
    @RequestGuid     uniqueidentifier = NULL OUTPUT
);
```

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

@EntityId: The identifier for the Entity of the list (1).

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

Return values: An integer that MUST be zero.

Result sets: MUST NOT return any result sets.

3.1.4.96 proc_TranExtendLockWeb

The **proc_TranExtendLockWeb** stored procedure is called to increase the duration of the short-term transaction application lock on a site (2). The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_TranExtendLockWeb (
    @WebId            uniqueidentifier,
    @WebLockDurationTime int,
    @TranLockerId     uniqueidentifier
);
```

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

@WebLockDurationTime: A value in seconds which the duration of the short-term transaction application lock MUST be increased by.

@TranLockerId: The identifier of the short-term transaction application lock for the current request.

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

Value	Description
0	Successful Execution.
158	The site (2) does not have a short-term transaction application lock set on it.
288	The site (2) already has a short-term transaction application lock and the @TranLockerId does not match the identifier of the short-term transaction application lock on the site.
1359	An internal error has occurred.
1931	The site (2) already has a short-term transaction application lock and all of the following are true:

Value	Description
	<ul style="list-style-type: none"> ▪ The @TranLockerId identifier matches the identifier of the short-term transaction application lock for the site (2). ▪ The current database time is greater than the expiry time of the short-term transaction application lock.

Result sets: MUST NOT return any result sets.

3.1.4.97 proc_TransLockWeb

The **proc_TransLockWeb** stored procedure is called to set a short-term transaction application lock on a site (2). The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_TransLockWeb (
    @WebId                uniqueidentifier,
    @WebLockDurationTime int,
    @TranLockerId        uniqueidentifier
);
```

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

@WebLockDurationTime: The duration in seconds for which to apply the short-term transaction application lock for the site (2).

@TranLockerId: The identifier of the short-term transaction application lock for the current request.

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

Value	Description
0	Successful execution.
212	The site (2) already has a short-term transaction application lock and all of the following are true: <ul style="list-style-type: none"> ▪ The @TranLockerId does not match the identifier of the short-term transaction application lock for the site. ▪ The current database time is less than the expiry time of the short-term transaction application lock.
1359	An internal error has occurred.

Result sets: MUST NOT return any result sets.

3.1.4.98 proc_TransUnlockWeb

The **proc_TransUnlockWeb** stored procedure is called to remove the short-term transaction application lock on a site (2). The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_TransUnlockWeb (
    @WebId                uniqueidentifier,
```

```

    @TranLockerId    uniqueidentifier
);

```

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

@TranLockerId: The identifier of the short-term transaction application lock for the current request.

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

Value	Description
0	Successful Execution.
158	The site (2) does not have a short-term transaction application lock set on it.
288	The site (2) already has a short-term transaction application lock and the @TranLockerId does not match the identifier of the short-term transaction application lock on the site (2).
1359	An internal error has occurred.
1931	The site (2) already has a short-term transaction application lock and all of the following are true: <ul style="list-style-type: none"> ▪ The @TranLockerId identifier matches the identifier of the short-term transaction application lock for the site. ▪ The current database time is greater than the expiry time of the short-term transaction application lock.

Result sets: MUST NOT return any result sets.

3.1.4.99 proc_UpdateDirtyDocument

The **proc_UpdateDirtyDocument** stored procedure is called to store changes to an item that is a folder or document. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_UpdateDirtyDocument (
    @DocSiteId    uniqueidentifier,
    @DocDirName   nvarchar(256),
    @DocLeafName  nvarchar(128),
    @Level        tinyint,
    @SendingContent bit,
    @DocSize      int,
    @CacheParseId uniqueidentifier,
    @Dynamic      bit,
    @@DocWebId    uniqueidentifier OUTPUT,
    @@DocId       uniqueidentifier OUTPUT,
    @@DoclibRowId int OUTPUT,
    @@DocDTW      datetime OUTPUT,
    @@DocVersion  int OUTPUT,
    @@DocUnghosted bit OUTPUT,
    @RequestGuid  uniqueidentifier = NULL OUTPUT
);

```

@DocSiteId: The site collection identifier containing the document to be updated. MUST NOT be NULL.

@DocDirName: The directory name component of the URL for the item to be updated.

@DocLeafName: The leaf name component of the URL for the item to be updated.

@Level: The publishing level of the item to update. Valid values are in [\[MS-WSSFO2\]](#) section 2.2.2.6.

@SendingContent: MUST be "1" if the document stream of the document is intended to be stored in the back-end database server. Otherwise, it MUST be zero.

@DocSize: The size, in bytes, of the document stream of the document if the document stream of the document is intended to be stored in the back-end database server. Otherwise, it MUST be zero.

@CacheParseId: A GUID designating the cached version of the item to update.

@Dynamic: If the value is 1, this is a dynamic page; otherwise it is a **static page**.

@@DocWebId: An output variable providing the site identifier containing the item updated.

@@DocId: An output variable providing the document identifier (2) of the item updated.

@@DoclibRowId: An output variable providing the row identifier of the item within the containing document library or list (1), if applicable.

@@DocDTW: An output variable providing a timestamp in Coordinated Universal Time (UTC) specifying when the last changes were made to the document stream. If the **@SendingContent** parameter is 1, **@@DocDTW** MUST be the date and time that this stored procedure is called.

@@DocVersion: An output variable providing an incremented version of the item.

@@DocUnghosted: An output variable that, if set to "1", indicates the item to be updated was uncustomized and is now customized (1).

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

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

Result sets: MUST NOT return any result sets.

3.1.4.100 **proc_UpdateItemInNameValuePair**

The **proc_UpdateItemInNameValuePair** stored procedure is called to update the indexed fields and their values of the specified list item. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_UpdateItemInNameValuePair(
    @SiteId                uniqueidentifier,
    @WebId                 uniqueidentifier,
    @ListId                uniqueidentifier,
    @ItemId                int,
    @Level                 tinyint = 1,
    @FieldId1              uniqueidentifier = NULL,
    @FieldValue1           sql_variant = NULL,
    @FieldId2              uniqueidentifier = NULL,
    @FieldValue2           sql_variant = NULL,
    @FieldId3              uniqueidentifier = NULL,
    @FieldValue3           sql_variant = NULL,
    @FieldId4              uniqueidentifier = NULL,
    @FieldValue4           sql_variant = NULL,
```

```

@FieldId5           uniqueidentifier = NULL,
@FieldValue5       sql_variant = NULL,
@FieldId6           uniqueidentifier = NULL,
@FieldValue6       sql_variant = NULL,
@FieldId7           uniqueidentifier = NULL,
@FieldValue7       sql_variant = NULL,
@FieldId8           uniqueidentifier = NULL,
@FieldValue8       sql_variant = NULL,
@FieldId9           uniqueidentifier = NULL,
@FieldValue9       sql_variant = NULL,
@FieldId10          uniqueidentifier = NULL,
@FieldValue10      sql_variant = NULL,
@InsertIfUpdateFails int = 0,
@SelectFromUserData bit = 0,
@RequestGuid        uniqueidentifier = NULL OUTPUT
);

```

@SiteId: The site collection identifier of the site collection that contains the specified list item.

@WebId: The site identifier of the site (2) which contains the specified list item.

@ListId: The list identifier of the list (1) which contains the specified list item.

@ItemId: The item identifier of the specified list item in the list (1).

@Level: The publishing level. The default value is 1.

@FieldId#: The field identifiers of the indexed fields. There are ten FieldId parameters numbered from 1 to 10. The default values are NULL.

@FieldValue#: The value of the indexed fields. There are ten FieldValue parameters numbered from 1 to 10. The default values are NULL.

@InsertIfUpdateFails: If list item is found, the indexed fields and values will be updated. If the value of this parameter is 1 and the specified list item cannot be found in the **NameValuePair** table (Section [2.2.7.2](#)), the indexed fields and values MUST be inserted in the **NameValuePair** table (Section [2.2.7.2](#)). If this parameter is not 1 and the specified list item cannot be found in the NameValuePair Table (Section [2.2.7.2](#)), the stored procedure MUST NOT insert the indexed fields and values into the **NameValuePair** table (Section [2.2.7.2](#)).

@SelectFromUserData: If this parameter is set to 1 and **@FieldId#** is not NULL and **@FieldId#** is one of the field identifier values in the following table, then the associated **@FieldValue#** is replaced with the corresponding column of the row in the **AllUserData** table (Section [2.2.7.1](#)) table specified by **@ListId**, **@ItemId**, and **@Level**. The default value is zero. This parameter MUST NOT be NULL.

Field Id	Field Name(Column in the AllUserData Table)
1df5e554-ec7e-46a6-901d-d85a3881cb18	tp_Author
d31655d1-1d5b-4511-95a1-7a09e9b75bf2	tp_Editor
8c06beca-0777-48f7-91c7-6da68bc07b69	tp_Created
28cf69c5-fa48-462a-b5cd-27b6f9d2bd5f	tp_Modified
26d0756c-986a-48a7-af35-bf18ab85ff4a	tp_HasCopyDestinations

Field Id	Field Name(Column in the AllUserData Table)
7841bf41-43d0-4434-9f50-a673baef7631	tp_UIVersion
03e45e84-1992-4d42-9116-26f756012634	tp_ContentTypeId
3881510a-4e4a-4ee8-b102-8ee8e2d0dd4b	tp_CheckoutUserId

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

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

Value	Description
0	Successful execution.
87	The input parameters are incorrect.

Result sets: MUST NOT return any result sets.

3.1.4.101 proc_UpdateItemInNameValuePairCollated

The **proc_UpdateItemInNameValuePairCollated** stored procedure is called to update the indexed fields and their values of the specified list item for a specific collation. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_UpdateItemInNameValuePairCollated(
    @SiteId                uniqueidentifier,
    @WebId                 uniqueidentifier,
    @ListId                uniqueidentifier,
    @ItemId                int,
    @Collation              smallint,
    @Level                 tinyint = 1,
    @FieldId1              uniqueidentifier = NULL,
    @FieldValue1           nvarchar(255) = NULL,
    @FieldId2              uniqueidentifier = NULL,
    @FieldValue2           nvarchar(255) = NULL,
    @FieldId3              uniqueidentifier = NULL,
    @FieldValue3           nvarchar(255) = NULL,
    @FieldId4              uniqueidentifier = NULL,
    @FieldValue4           nvarchar(255) = NULL,
    @FieldId5              uniqueidentifier = NULL,
    @FieldValue5           nvarchar(255) = NULL,
    @FieldId6              uniqueidentifier = NULL,
    @FieldValue6           nvarchar(255) = NULL,
    @FieldId7              uniqueidentifier = NULL,
    @FieldValue7           nvarchar(255) = NULL,
    @FieldId8              uniqueidentifier = NULL,
    @FieldValue8           nvarchar(255) = NULL,
    @FieldId9              uniqueidentifier = NULL,
    @FieldValue9           nvarchar(255) = NULL,
    @FieldId10             uniqueidentifier = NULL,
    @FieldValue10          nvarchar(255) = NULL,
    @InsertIfUpdateFails  int = 0
);

```

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

@WebId: The site identifier of the site (2) which contains the specified list item.

@ListId: The list identifier of the list (1) which contains the specified list item.

@ItemId: The item identifier of the specified list item in the list (1).

@Collation: The collation identifier of the collation for the specified list item in the list (1).

@Level: The publishing level. The default value is 1.

@FieldId#: The field identifier of the indexed fields. There are ten **FieldId** parameters numbered from 1 to 10. The default values are NULL.

@FieldValue#: The value of the indexed fields. There are ten **FieldValue** parameters numbered from 1 to 10. The default values are NULL.

@InsertIfUpdateFails: If this parameter value is 1, the indexed fields and values will be inserted into the **NameValuePair** table (Section [2.2.7.2](#)), if the specified list item cannot be found in the **NameValuePair** table. If list item is found, the indexed fields and values will be updated. If the value of this parameter is not 1 and the list item is not found, the indexed fields and values will not be inserted into the **NameValuePair** table. The default value is zero.

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

Value	Description
0	Successful execution.
87	The input parameters are not valid.

Result sets: MUST NOT return any result sets.

3.1.4.102 **proc_UpdateLookupRelationship**

The **proc_UpdateLookupRelationship** stored procedure is called to update the relationship delete behavior on a relationship lookup field in a list (1). The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_UpdateLookupRelationship (  
    @SiteId uniqueidentifier,  
    @WebId uniqueidentifier,  
    @ListId uniqueidentifier,  
    @FieldId uniqueidentifier,  
    @LookupListId uniqueidentifier,  
    @DeleteBehavior tinyint,  
    @RequestId uniqueidentifier = null OUTPUT);
```

@SiteId: The site collection identifier of the site collection which contains the specified list (1).

@WebId: The site identifier of the site (2) which contains the specified list (1).

@ListId: The list identifier of the list (1) containing the specified relationship lookup field.

@FieldId: The field identifier of the specified relationship lookup field.

@LookupListId: The list identifier of the target list (1) for the specified relationship lookup field.

@DeleteBehavior: A value which specifies the type of relationship delete behavior that MUST be set on the specified relationship lookup field. If the value is zero and a relationship delete behavior is set on the relationship lookup field, it MUST be removed. If the value is 1 or 2, the corresponding relationship delete behavior specified in [Relationship Delete Behavior Type](#) MUST be set.

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

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

Value	Description
0	Successful execution.
19	The target list specified by @LookupListId does not exist and @DeleteBehavior is either 1 or 2.
155	The list server template (as specified in [MS-WSSFO2] section 2.2.3.12) of the list specified by @ListId or the target list (1) specified by @LookupListId is 202 and @DeleteBehavior is either 1 or 2.
1359	An internal error occurred.

Result sets: MUST NOT return any result sets.

3.1.4.103 proc_UpdateOrderNumber

The **proc_UpdateOrderNumber** stored procedure is called to update the value of the **item order field (1)** in an existing list item in order for the list item to be displayed at the specified position when a set of list items in the list (1) are sorted by the item order field in ascending order. A set of list items could be all list items in the list (1), or all list items that are associated with a particular **meeting instance** in the list (1) or all list items that are under a folder in the list (1). Item order field is a field (1) with name "Order" and GUID {ca4addac-796f-4b23-b093-d2a3f65c0774}. When a list (1) has the item order field and the set of list items are sorted by the item order field in ascending order, the value of the item order field determines the displaying position of the list item. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_UpdateOrderNumber (
    @SiteId                uniqueidentifier,
    @WebId                 uniqueidentifier
    @ListId                uniqueidentifier,
    @BaseType              int,
    @ItemId                int,
    @ItemOrder             int,
    @fMultipleMtgDataList bit,
    @RootFolderUrl         nvarchar(256),
    @fUpdateDTM            bit = 0,
    @InstanceId            int = -3,
    @RequestGuid           uniqueidentifier = NULL OUTPUT
);
```

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

@WebId: The site identifier of the site (2) which contains the specified list item.

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

@BaseType: The list (1) base type of the list (1) that contains the list item to be updated. See [\[MS-WSSFO2\]](#), section [2.2.3.11](#).

@ItemId: The item identifier of the list item to be updated.

@ItemOrder: The displaying position of the list item when the set of list items are sorted by the item order field in ascending order. The first displaying position is 1, the second displaying position is 2, and so on. If **@ItemOrder** is less than 1, it is the same as the first displaying position. If the **@ItemOrder** is greater than the total number of the set of list items, it is same as the last displaying position. The **@ItemOrder** can be NULL. When **@ItemOrder** is NULL, the stored procedure MUST use **@ItemId** times 100 as the value for item order field in the list item.

@fMultipleMtgDataList: A bit flag specifying whether the site (2) specified by **@WebId** is a Meeting Workspace site and the list (1) specified by **@ListId** contains data for multiple meeting instances of a recurring meeting within the site. The **@fMultipleMtgDataList** MUST NOT be NULL. When a meeting is a recurring meeting, there are multiple meeting instances and each meeting instance has its own integer identifier. When **@fMultipleMtgDataList** is 1, the set of list items are all list items that are associated with the meeting instance specified by **@InstanceId**.

@RootFolderUrl: The directory name under which the set of list items are stored. If **@BaseType** is 1, **@RootFolderUrl** MUST NOT be NULL and the set of list items are all list items stored under the directory name. If **@BaseType** is not 1, this parameter MUST be NULL.

@fUpdateDTM: A bit flag specifying whether to update the list (1)'s last modified date time. It MUST NOT be NULL. The default value is zero. When its value is "1", the stored procedure MUST update the last modified date time of the list (1) using the current Coordinated Universal Time (UTC) datetime.

@InstanceId: The integer identifier of a meeting instance in a recurring meeting. When a meeting is a recurring meeting, there are multiple meeting instances. **@InstanceId** is the integer identifier of a meeting instance in the recurring meeting. It MUST NOT be NULL. The default value is -3. When its value is -3 and **@fMultipleMtgDataList** is 1, the stored procedure MUST use the integer identifier of the meeting instance associated with the list item as the meeting instance identifier.

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

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

Value	Description
0	Successful execution.
13	The proc_UpdateOrderNumber cannot determine a value for item order field to ensure the list item be displayed at the specified position.

Result sets: MUST NOT return any result sets.

3.1.4.104 proc_UpdateVersionVirusInfo

The **proc_UpdateVersionVirusInfo** stored procedure is called to update the latest virus scanning related information for the specified version of the document. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_UpdateVersionVirusInfo (  
    @DocSiteId          uniqueidentifier,  
    @DocId              uniqueidentifier,
```

```

    @Version          int,
    @VirusVendorID   int,
    @VirusStatus     int,
    @VirusInfo       nvarchar(255),
    @RequestGuid     uniqueidentifier = NULL OUTPUT
);

```

@DocSiteId: The site collection identifier of the site collection containing the document version to be updated.

@DocId: The document identifier (2) of the document version.

@Version: UI version number for the document.

@VirusVendorID: The identifier of the virus scanner that processed this document. This value MUST be NULL if this document has not been processed by a virus scanner.

@VirusStatus: An enumerated type specifying the current virus state of this document. This value MUST be NULL if it has not been processed by a virus scanner. Valid values are listed in [\[MS-WSSFO2\]](#) section 2.2.3.17.

@VirusInfo: A string containing a provider specific message returned by the virus scanner when it last processed the document. This value MUST be NULL if it has not been processed by a virus scanner.

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

Return values: An integer that MUST be zero.

Result sets: MUST NOT return any result sets.

3.1.4.105 `proc_UpdateView`

The **proc_UpdateView** stored procedure is called to save modifications to the specified view. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_UpdateView(
    @SiteId          uniqueidentifier,
    @ListId          uniqueidentifier,
    @ViewId          uniqueidentifier,
    @UserId          int,
    @IsPersonalView bit,
    @View            nvarchar(max),
    @DisplayName     nvarchar(255),
    @ContentTypeId  varbinary(512),
    @ViewFlags      int,
    @ViewMask       int,
    @Level           tinyint,
    @BypassCheck    bit = 0,
    @bUpdateAllPersonalViews bit = 0
);

```

@SiteId: The site collection identifier of the site collection which contains the specified view for the specified list (1).

@ListId: The list identifier of the list (1) for the specified view.

@ViewId: The view identifier of the view for which modifications will be saved.

@UserId: The user identifier that originally created the specified personal view. This parameter MUST be NULL for shared views.

@IsPersonalView: This parameter MUST be one of the values in the following table.

Value	Description
0	The stored procedure updates the shared view specified by the @ListId parameter and the @ViewId parameter.
1	The stored procedure updates the personal view specified by the @ListId parameter and the @ViewId parameter.

@View: A **query** expressed in Collaborative Application Markup Language (CAML) and converted into a compressed structure as described in [\[MS-WSSFO2\]](#) section 2.2.4.8. See [\[MS-WSSCAML\]](#) section 2.2 for more information about the Collaborative Application Markup Language (CAML). If this parameter is NULL, then the existing CAML query is not modified.

@DisplayName: The name for the newly created view.

@ContentTypeId: The new content type identifier for the specified view. If this parameter is NULL, then the existing content type identifier for the specified view is not modified.

@ViewFlags: The new view flags for the specified view. If this parameter is NULL, then the existing view flags are not modified and **@ViewMask** MUST be ignored.

@ViewMask: The view flags that will be removed from the existing set of view flags when applying the **@ViewFlags**. If the **@ViewFlags** parameter is not NULL, this parameter MUST NOT be NULL.

@Level: The publishing level for the view specified by the **@ViewId** parameter.

@BypassCheck: If this parameter is zero then the following validation steps MUST occur. If this parameter is 1, then the following validation steps MUST NOT occur.

- If the view specified by **@ViewId** has been deleted then the view MUST not be modified.
- If **@IsPersonalView** is set to zero and the document version of the view specified is not the most recent version, then the view MUST NOT be modified.
- If **@IsPersonalView** is set to zero and the list (1) specified by **@ListId** requires list items to be checked out prior to editing them and the publishing level for the view's document is not checked out, then the view MUST not be modified.

@bUpdateAllPersonalViews: If the parameter is set to 1, the stored procedure will modify the personal view that is specified by **@ViewId** regardless of the **@UserId**.

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

Value	Description
-2147467259	The modifications for the specified view were not saved because an error occurred while updating the properties of the Web Part associated with the view.
0	Successful execution.

Value	Description
1	The modifications for the specified view were not saved because of an error in the back-end database server.
3	The modifications for the specified view were not saved because one of the following: <ul style="list-style-type: none"> ▪ The @Level parameter supplied was NULL. ▪ The @BypassCheck parameter is zero and the document specified by the @ViewId parameter has been deleted. ▪ The view specified by the @ViewId parameter with the specified @UserId and @Level and @IsPersonalView for the list (1) specified by the @listId parameter does not exist.
33	The modifications for the specified view were not saved because all of the following conditions apply: <ul style="list-style-type: none"> ▪ The @BypassCheck parameter is zero. ▪ The @IsPersonalView parameter is zero. ▪ The document specified by the @ViewId parameter is not the most recent document version.
158	The modifications for the specified view were not saved because all of the following conditions apply: <ul style="list-style-type: none"> ▪ The @BypassCheck parameter was zero. ▪ The @IsPersonalView parameter was zero. ▪ The document specified by the @ViewId parameter is not checked out. ▪ The list (1) specified by the @ListId parameter mandates that documents MUST be checked out prior to making document modifications.
212	The specified view was not modified because the site collection has its WRITELOCK (0x00000001) site collection flag bit set.
1816	The specified view was not modified because saving the modifications for the view would have exceeded the site collection quota.

Result sets: MUST NOT return any result sets.

3.1.4.106 **proc_UpdateVirusInfo**

The **proc_UpdateVirusInfo** stored procedure is called to update the latest virus scanning information for the current version of the document. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_UpdateVirusInfo(
    @DocSiteId        uniqueidentifier,
    @DocId            uniqueidentifier,
    @DocLevel         tinyint,
    @DocVersion       int,
    @VirusVendorID    int,

```

```

    @VirusStatus      int,
    @VirusInfo        nvarchar(255),
    @SendingContent   bit,
    @DocSize          int,
    @RequestGuid      uniqueidentifier = NULL OUTPUT
);

```

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

@DocId: The document identifier (2) of the specified document.

@DocLevel: The publishing level of the specified document.

@DocVersion: The internal version number of the specified document.

@VirusVendorID: The identifier of the virus scanner that processed the specified document. This value MUST be NULL if this document has not been processed by a virus scanner.

@VirusStatus: An enumerated type specifying the current virus state of this document. This value MUST be NULL if it has not been processed by a virus scanner. Valid values are listed in [\[MS-WSSFO2\]](#) section 2.2.3.17.

@VirusInfo: A string containing a provider specific message returned by the virus scanner when it last processed the document. This value MUST be NULL if it has not been processed by a virus scanner.

@SendingContent: MUST be 1 if the document stream of the document is intended to be stored in the back end database server; otherwise it MUST be zero.

@DocSize: Size, in bytes, of the document.

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

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

Value	Description
0	Successful execution.
3	Document not found or @SendingContent is 1 and the document stream for the document was not found.

Result sets: MUST NOT return any result sets.

3.1.4.107 proc_UpdateWebPartLinks

The **proc_UpdateWebPartLinks** stored procedure is called during link fixup to update Web Part properties. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_UpdateWebPartLinks (
    @SiteId            uniqueidentifier,
    @WebPartID        uniqueidentifier,
    @Level            tinyint,
    @AllUsersProperties varbinary(max),
    @Source            nvarchar(max),
    @RequestGuid      uniqueidentifier = NULL OUTPUT
)

```

);

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

@WebPartID: The GUID of the Web Part for which to update Web Part properties. MUST NOT be NULL.

@Level: Indicates the publishing level of the **Web Part Page** containing the Web Part.

@AllUsersProperties: A serialized representation of zero or more customizable properties on the Web Part. If this value is NULL, then default values will be used for all of the customizable properties on the Web Part.

@Source: Serialized representation of zero or more properties that can be personalized of a Web Part in an alternative format used by an HTML editor.

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

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

Value	Description
0	Successful execution.
2	The Web Part specified by @SiteId , @WebPartID , and @Level cannot be found.
212	The specified site collection is locked.
1359	Internal Error.
1816	The quota for the specified site collection has been exceeded.
-2147467259	Unknown error.

Result sets: MUST NOT return any result sets.

3.1.4.108 **proc_UserHasDataItems**

The **proc_UserHasDataItems** stored procedure is called to determine whether a particular user has created any list items in the specified list (1). The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_UserHasDataItems (  
    @ListID          uniqueidentifier,  
    @UserID          int,  
    @InstanceID     int,  
    @RequestGuid    uniqueidentifier = NULL OUTPUT  
);
```

@ListID: The list identifier of the list (1). This parameter MUST NOT be NULL.

@UserID: The identifier of the user.

@InstanceID: If this list is inside a Meeting Workspace site, the identifier of the meeting instance of the list (1) that will be checked for list items. If the specified list (1) is not associated with a meeting instance, the **@InstanceID** MUST be equal to NULL.

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

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

Value	Description
0	The user has not created any list items or the specified @ListID does not exist or the specified @UserID does not exist.
1	The user has at least one list item created in the specified list (1).

Result sets: MUST NOT return any result sets.

3.1.4.109 proc_ValidateLookupParents

The **proc_ValidateLookupParents** stored procedure is called to verify whether there exist list items for specified list item identifiers in target lists (1) of specified relationship lookup fields of a specified list (1). The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ValidateLookupParents (
    @SiteId                uniqueidentifier,
    @ListId                uniqueidentifier,
    @IsInsert              bit,
    @ItemId                int,
    @LookupFieldId1        uniqueidentifier = null,
    @ParentId1            int = null,
    @IsInvalid1            bit = null OUTPUT,
    @LookupFieldId2        uniqueidentifier = null,
    @ParentId2            int = null,
    @IsInvalid2            bit = null OUTPUT,
    @LookupFieldId3        uniqueidentifier = null,
    @ParentId3            int = null,
    @IsInvalid3            bit = null OUTPUT,
    @LookupFieldId4        uniqueidentifier = null,
    @ParentId4            int = null,
    @IsInvalid4            bit = null OUTPUT,
    @LookupFieldId5        uniqueidentifier = null,
    @ParentId5            int = null,
    @IsInvalid5            bit = null OUTPUT,
    @LookupFieldId6        uniqueidentifier = null,
    @ParentId6            int = null,
    @IsInvalid6            bit = null OUTPUT,
    @LookupFieldId7        uniqueidentifier = null,
    @ParentId7            int = null,
    @IsInvalid7            bit = null OUTPUT,
    @LookupFieldId8        uniqueidentifier = null,
    @ParentId8            int = null,
    @IsInvalid8            bit = null OUTPUT,
    @LookupFieldId9        uniqueidentifier = null,
    @ParentId9            int = null,
    @IsInvalid9            bit = null OUTPUT,
    @LookupFieldId10       uniqueidentifier = null,
    @ParentId10           int = null,
    @IsInvalid10           bit = null OUTPUT,
    @RequestGuid           uniqueidentifier = null OUTPUT
);
```


@SiteId: The site collection identifier of the site collection that contains the specified list (1).

@ListId: The list identifier of the specified list (1).

@IsInsert: A bit flag specifying whether the stored procedure is called as part of the insertion or deletion of a list item, document or folder. If this parameter is set to 1 then the list item, document or folder is being inserted. If this parameter is set to zero then the list item, document or folder is being updated.

@ItemId: The list item identifier of the list item which is being inserted into the specified list (1) or updated in the transaction which calls the stored procedure.

@LookupFieldId#: The field identifier of the specified relationship lookup field. There are 10 specified relationship lookup fields numbered from 1 to 10. It can be NULL.

@ParentId#: The specified list item identifier to verify for existence in the target list (1) of the corresponding numbered relationship lookup field. There are 10 provided list item identifiers corresponding to each relationship lookup field numbered from 1 to 10. If the corresponding @LookupFieldId# is NULL then this parameter MUST be NULL.

@IsValid#: A bit flag which is an output parameter. There are 10 provided bit flags corresponding to each relationship lookup field numbered from 1 to 10 and each MUST specify a valid value from the following table.

Value	Description
0	Either of the following conditions apply: <ul style="list-style-type: none">▪ The corresponding @ParentId# and @LookupFieldId# values are NULL.▪ There exists a list item with the list item identifier specified by the corresponding @ParentId# in the target list (1) of the relationship lookup field specified by @LookupFieldId#.
1	Both of the following conditions apply: <ul style="list-style-type: none">▪ The corresponding @ParentId# and @LookupField# values are not NULL.▪ There is no list item with the list item identifier specified by the corresponding @ParentId# in the target list (1) of the relationship lookup field specified by @LookupFieldId#.

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

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

Value	Description
0	Either of the following conditions apply for each of the provided relationship lookup fields specified by @FieldId# and list item identifiers specified by @ParentId#: <ul style="list-style-type: none">▪ The ParentId# and @LookupField# values are NULL.▪ There exists a list item with the list item identifier specified by the corresponding @ParentId# in the target list (1) of the relationship lookup field specified by @LookupFieldId#.

Value	Description
6	<p>Both of the following is true for at least one of the provided relationship lookup fields specified by @FieldId# and list item identifiers specified by @ParentId#:</p> <ul style="list-style-type: none"> ▪ The ParentId# and @LookupField# values are not NULL. ▪ There is no list item with the list item identifier specified by the corresponding @ParentId# in the target list (1) of the relationship lookup field specified by @LookupFieldId#.

Result sets: MUST NOT return any result sets.

3.1.4.110 **proc_ValidateUniqueFields**

The **proc_ValidateUniqueFields** stored procedure is called to verify whether there is a list item other than a specified list item in the list (1) which has the same values as the ones provided for a given set of fields. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_ValidateUniqueFields (
    @SiteId                uniqueidentifier,
    @ListId                 uniqueidentifier,
    @ItemId                 int,
    @FieldId1               uniqueidentifier = null,
    @FieldValue1            sql_variant = null,
    @HasDuplicates1         bit = null OUTPUT,
    @FieldId2               uniqueidentifier = null,
    @FieldValue2            sql_variant = null,
    @HasDuplicates2         bit = null OUTPUT,
    @FieldId3               uniqueidentifier = null,
    @FieldValue3            sql_variant = null,
    @HasDuplicates3         bit = null OUTPUT,
    @FieldId4               uniqueidentifier = null,
    @FieldValue4            sql_variant = null,
    @HasDuplicates4         bit = null OUTPUT,
    @FieldId5               uniqueidentifier = null,
    @FieldValue5            sql_variant = null,
    @HasDuplicates5         bit = null OUTPUT,
    @FieldId6               uniqueidentifier = null,
    @FieldValue6            sql_variant = null,
    @HasDuplicates6         bit = null OUTPUT,
    @FieldId7               uniqueidentifier = null,
    @FieldValue7            sql_variant = null,
    @HasDuplicates7         bit = null OUTPUT,
    @FieldId8               uniqueidentifier = null,
    @FieldValue8            sql_variant = null,
    @HasDuplicates8         bit = null OUTPUT,
    @FieldId9               uniqueidentifier = null,
    @FieldValue9            sql_variant = null,
    @HasDuplicates9         bit = null OUTPUT,
    @FieldId10              uniqueidentifier = null,
    @FieldValue10           sql_variant = null,
    @HasDuplicates10        bit = null OUTPUT,
    @RequestGuid            uniqueidentifier = null OUTPUT
);

```

@SiteId: The site collection identifier of the site collection which contains the specified list (1).

@ListId: The list identifier of the specified list (1).

@ItemId: The list item identifier of the specified list item.

@FieldId#: The field identifier of the specified field (1). There are 10 specified fields (1) numbered from 1 to 10. If fewer than 10 fields are being validated, the remaining **@FieldId#** values MUST be NULL.

@FieldValue#: The provided value of the specified field (1). There are 10 provided values corresponding to each field (1) numbered from 1 to 10. If the corresponding **@FieldId#** is NULL then this parameter MUST be NULL.

@HasDuplicates#: A bit flag which is an output parameter. There are 10 such bit flags corresponding to each of the fields (1) numbered from 1 to 10 and each MUST specify a valid value from the following table.

Value	Description
0	Either of the following conditions are true: <ul style="list-style-type: none">▪ The corresponding @FieldId# and @FieldValue# values are NULL or empty.▪ The list (1) contains no list item other than the specified list item with the value specified by the corresponding @FieldValue# for the field (1) specified by the corresponding @FieldId#.
1	Both of the following conditions are true: <ul style="list-style-type: none">▪ The corresponding @FieldId# and @FieldValue# values are not NULL or empty.▪ The list (1) contains at least one list item other than the specified list item with the value specified by the corresponding @FieldValue# for the field (1) specified by the corresponding @FieldId#.

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

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

Value	Description
0	Either of the following is true for each of the provided fields specified by @FieldId# and values specified by @FieldValue#: <ul style="list-style-type: none">▪ The corresponding @FieldId# and @FieldValue# values are NULL or empty.▪ The list (1) contains no list item other than the specified list item with the value specified by the corresponding @FieldValue# for the field (1) specified by the corresponding @FieldId#.
2014	Both of the following is true for at least one of the provided fields (1) specified by @FieldId# and values specified by @FieldValue#: <ul style="list-style-type: none">▪ The corresponding @FieldId# and @FieldValue# values are not NULL or empty.▪ The list (1) contains at least one list item other than the specified list item with the value specified by the corresponding @FieldValue# for the field (1) specified by the corresponding @FieldId#.

Result sets: MUST NOT return any result sets.

3.1.4.111 **proc_ValidateUniqueFieldsCollated**

The **proc_ValidateUniqueFieldsCollated** stored procedure is called to verify whether there is a list item other than a specified list item in the list (1) which has the same values, as defined by the specified collation order, as the ones provided for a given set of fields. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ValidateUniqueFieldsCollated (
    @SiteId                uniqueidentifier,
    @ListId                uniqueidentifier,
    @ItemId                int,
    @Collation              smallint,
    @FieldId1              uniqueidentifier = null,
    @FieldValue1           nvarchar(255) = null,
    @HasDuplicates1       bit = null OUTPUT,
    @FieldId2              uniqueidentifier = null,
    @FieldValue2           nvarchar(255) = null,
    @HasDuplicates2       bit = null OUTPUT,
    @FieldId3              uniqueidentifier = null,
    @FieldValue3           nvarchar(255) = null,
    @HasDuplicates3       bit = null OUTPUT,
    @FieldId4              uniqueidentifier = null,
    @FieldValue4           nvarchar(255) = null,
    @HasDuplicates4       bit = null OUTPUT,
    @FieldId5              uniqueidentifier = null,
    @FieldValue5           nvarchar(255) = null,
    @HasDuplicates5       bit = null OUTPUT,
    @FieldId6              uniqueidentifier = null,
    @FieldValue6           nvarchar(255) = null,
    @HasDuplicates6       bit = null OUTPUT,
    @FieldId7              uniqueidentifier = null,
    @FieldValue7           nvarchar(255) = null,
    @HasDuplicates7       bit = null OUTPUT,
    @FieldId8              uniqueidentifier = null,
    @FieldValue8           nvarchar(255) = null,
    @HasDuplicates8       bit = null OUTPUT,
    @FieldId9              uniqueidentifier = null,
    @FieldValue9           nvarchar(255) = null,
    @HasDuplicates9       bit = null OUTPUT,
    @FieldId10             uniqueidentifier = null,
    @FieldValue10          nvarchar(255) = null,
    @HasDuplicates10      bit = null OUTPUT,
    @RequestGuid           uniqueidentifier = null OUTPUT
);
```

@SiteId: The site collection identifier of the site collection which contains the specified list (1).

@ListId: The list identifier of the specified list (1).

@ItemId: The list item identifier of the specified list item.

@Collation: The collation order for the site (2) that contains the list (1). It MUST NOT be NULL and MUST be one of the values defined in Collation Order Enumeration, [\[MS-WSSFO2\]](#) section 2.2.3.4.

@FieldId#: The field identifier of the specified field (1). There are 10 specified fields numbered from 1 to 10. If the field (1) does not contain data, it MUST be NULL.

@FieldValue#: The provided value of the specified field (1). There are 10 provided values corresponding to each field (1) numbered from 1 to 10. If the provided value does not contain data, it MUST be NULL.

@HasDuplicates#: A bit flag which is an output parameter. There are 10 such bit flags corresponding to each of the fields (1) numbered from 1 to 10 and it MUST specify a valid value from the following table.

Value	Description
0	<p>Either of the following conditions are true:</p> <ul style="list-style-type: none"> The corresponding @FieldId# and @FieldValue# values are NULL or empty. The list (1) contains no list item other than the specified list item with the value specified by the corresponding @FieldValue# for the field (1) specified by the corresponding @FieldId# where equality is defined by the specified collation order.
1	<p>Both of the following conditions are true:</p> <ul style="list-style-type: none"> The corresponding @FieldId# and @FieldValue# values are not NULL or empty. The list (1) contains at least one list item other than the specified list item with the value specified by the corresponding @FieldValue# for the field (1) specified by the corresponding @FieldId# where equality is defined by the specified collation order.

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

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

Value	Description
0	<p>Either of the following conditions apply for each of the provided fields specified by @FieldId# and values specified by @FieldValue#:</p> <ul style="list-style-type: none"> The corresponding @FieldId# and @FieldValue# values are NULL or empty. The list (1) contains no list item other than the specified list item with the value specified by the corresponding @FieldValue# for the field (1) specified by the corresponding @FieldId# where equality is defined by the specified collation order.
2014	<p>Both of the following conditions apply for at least one of the provided fields specified by @FieldId# and values specified by @FieldValue#:</p> <ul style="list-style-type: none"> The corresponding @FieldId# and @FieldValue# values are not NULL or empty. The list (1) contains at least one list item other than the specified list item with the value specified by the corresponding @FieldValue# for the field (1) specified by the corresponding @FieldId# where equality is defined by the specified collation order.

Result sets: MUST NOT return any result sets.

3.1.4.112 proc_VerifyListItemsTreeToDelete

The **proc_VerifyListItemsTreeToDelete** stored procedure is called to compare the child items to be deleted as part of a cascading delete operation for a specified parent item with a specified collection of child items. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_VerifyListItemsTreeToDelete (
    @ISiteId          uniqueidentifier,
    @IWebId           uniqueidentifier,
    @IListId          uniqueidentifier,
    @IItemId          int,
    @MaxLimit         int,
    @CsvListItemString nvarchar(max)
);
```

@ISiteId: The site collection identifier of the site collection containing the specified parent item of the cascading delete operation.

@IWebId: The site identifier of the site (2) containing the specified parent item.

@IListId: The list identifier of the list (1) containing the specified parent item.

@IItemId: The list item identifier of the specified parent item.

@MaxLimit: A value which specifies the number of child items allowed to be part of the cascading delete operation.

@CsvListItemString: A parameter specifying the set of child items to be deleted as part of the cascading delete operation. This parameter **MUST** conform to the following ABNF:

CsvListItemString = ListAndListItems 0*("," \$ "," ListAndListItems)

ListAndListItems = ListId 1*("," ListItemId)

ListId = GUID

ListItemId = 1*DIGIT

GUID = "{" GUIDCORE "}" / GUIDCORE

GUIDCORE = 8HEXDIG "-" 4HEXDIG "-" 4HEXDIG "-" 4HEXDIG "-" 12HEXDIG

The ListId **MUST** specify the list identifier containing the child item. The ListItemId **MUST** specify the list item identifier of the child item and it **MUST** be a 32-bit integer greater than zero.

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

Value	Description
0	Successful execution.
255	The set of child items specified by @CsvListItemString is different from the set of child items calculated for the specified parent item of the cascading delete operation, which is calculated as in List Items Tree Result Set (section 3.1.4.60.1).
1142	The number of child items of the cascading delete operation found is equal to or greater than @MaxLimit , which is calculated as in List Items Tree Result Set .

Value	Description
8239	<p>All of the following conditions apply for at least one list item, document or folder and at least one relationship lookup field of the list (1) containing it:</p> <ul style="list-style-type: none"> ▪ The relationship lookup field has the restrict behavior set on it. ▪ Either of the following conditions applies: <ul style="list-style-type: none"> ▪ The target list (1) of the relationship lookup field is the list (1) containing a child item of the cascading delete operation and either of the following is true for its value for the relationship lookup field in its most recent published or draft version: <ul style="list-style-type: none"> ▪ It is the list item identifier of that child item. ▪ It is the list item identifier of a list item, document or folder contained in that child item if it is a folder. ▪ The target list (1) of the relationship lookup field is the list (1) containing the parent item of the cascading delete behavior and either of the following is true for its value for the relationship lookup field in its most recent published or draft version: <ul style="list-style-type: none"> ▪ It is the list item identifier of the parent item. ▪ It is the list item identifier of a list item, document or folder contained in the parent item if it is a folder.

Result sets: MUST NOT return any result sets.

3.1.4.113 **proc_DeleteFileFragmentsById**

The **proc_DeleteFileFragmentsById** stored procedure is called to delete the file fragments for a document based on the value of a file fragment identifier. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_DeleteFileFragmentsById (
    @SiteId                uniqueidentifier,
    @DocId                 uniqueidentifier,
    @DocVersion            int,
    @ContentVersion        int,
    @UserId                int,
    @Partition             tinyint,
    @Id                    bigint,
    @DeleteUntil           bit,
    @DeleteOnly            bit,
    @CheckPerms            bit,
    @UpdateQuota           bit,
    @QuotaChange           bigint OUTPUT
);

```

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

@DocId: The document identifier (2) of the document associated with the file fragments being deleted.

@DocVersion: The current internal version number of the document to update. This parameter MUST be ignored if **@CheckPerms** is not 1.

@ContentVersion: The current content version number of the document to update. This parameter MUST be ignored if **@CheckPerms** is not 1.

@UserId: The user identifier of the current user making the request. This parameter MUST be ignored if **@CheckPerms** is not 1.

@Partition: The identifier for a file fragment partition of the file fragment partition where the file fragments to delete are located.

@Id: The file fragment identifier manipulated by this stored procedure. This parameter MUST be ignored if **@DeleteUntil** is not 1 and **@DeleteOnly** is not 1.

@DeleteUntil: If this parameter is 1, **@DeleteOnly** MUST NOT be 1 and file fragments with file fragment identifier less than **@Id** MUST be deleted. Else if **@DeleteOnly** is not 1, all file fragments MUST be deleted.

@DeleteOnly: If this parameter is 1, **@DeleteUntil** MUST NOT be 1 and the file fragment with file fragment identifier equal to **@Id** MUST be deleted. Else if **@DeleteUntil** is not 1, all file fragments MUST be deleted.

@CheckPerms: If this parameter is 1, **proc_FileFragmentPermissionCheck** MUST be called with parameters **@SiteId**, **@DocId**, **@UserId**, **@DocVersion**, and **@ContentVersion**.

@UpdateQuota: If this parameter is 1, the size in bytes of the file fragments deleted MUST be reflected in the quota.

@QuotaChange: The net amount of change to the file fragment data deleted, in bytes. This parameter MUST not be set to NULL.

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

Value	Description
0	Successful execution.
212	@UpdateQuota is 1 and the site collection is over quota or is write-locked.
4317	There were no file fragments deleted.
	Return codes from proc_FileFragmentPermissionCheck can be returned here.

Result sets: MUST NOT return any result sets.

3.1.4.114 **proc_DeleteFileFragmentsForCleanup**

The **proc_DeleteFileFragmentsForCleanup** stored procedure is called to delete file fragments for a specified group of documents. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_DeleteFileFragmentsForCleanup (
    @DocTypesToExclude    nvarchar(max),
    @NrDocsToFetchAndDelete int,
    @InLastDeletedDocId  uniqueidentifier,
    @NrDocsProcessed      int OUTPUT,
    @OutLastDeletedDocId uniqueidentifier OUTPUT
);
```


@DocTypesToExclude: A comma-separated string of file name extensions whose file fragments MUST not be deleted. This parameter MUST not be null.

@NrDocsToFetchAndDelete: Specifies the batch size. The procedure MUST not delete file fragments for more than this number of documents.

@InLastDeletedDocId: A document identifier (2) that MUST be the smallest document identifiers (2) for documents are updated in this execution.

@NrDocsProcessed: This number MUST be equal to the number of documents whose file fragments were deleted. This number MUST be less than or equal to **@NrDocsToFetchAndDelete**.

@OutLastDeletedDocId: The largest document identifier of the documents that were updated.

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

Value	Description
0	Successful execution.

Result sets: MUST NOT return any result sets.

3.1.4.115 **proc_DeleteFileFragmentsForUserDataDelete**

The **proc_DeleteFileFragmentsForUserDataDelete** stored procedure is called to delete file fragments for a specified group of documents. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_DeleteFileFragmentsForUserDataDelete (  
    @DocTypesToInclude      nvarchar(max),  
    @SizeThreshold          int,  
    @NrDocsToFetchAndDelete int,  
    @InLastDeletedDocId    uniqueidentifier,  
    @NrDocsProcessed       int OUTPUT,  
    @OutLastDeletedDocId   uniqueidentifier OUTPUT  
);
```

@DocTypesToInclude: A comma-separated string of file name extensions whose file fragments MUST be deleted. This parameter MUST not be null.

@SizeThreshold: Size threshold in bytes. The procedure MUST not update any documents whose total size of file fragments does not exceed this value.

@NrDocsToFetchAndDelete: Specifies the batch size. The procedure MUST not delete file fragments for more than this number of documents.

@InLastDeletedDocId: A document identifier (2) that MUST be the smallest document identifiers (2) for documents are updated in this execution.

@NrDocsProcessed: This number MUST be equal to the number of documents whose file fragments were deleted. This number MUST be less than or equal to **@NrDocsToFetchAndDelete**.

@OutLastDeletedDocId: The largest document identifier of the documents that were updated.

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

Value	Description
0	Successful execution.

Result sets: MUST NOT return any result sets.

3.1.4.116 `proc_GetDocStreamsForExternalMigration`

The `proc_GetDocStreamsForExternalMigration` stored procedure is called to migrate document streams to external storage. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_GetDocStreamsForExternalMigration (
    @ChunkSize          int,
    @PageSize           int,
    @FromSiteId         uniqueidentifier,
    @FromDeleteTransactionId varbinary(16),
    @FromParentId       uniqueidentifier,
    @FromId             uniqueidentifier,
    @FromLevel          tinyint
);

```

@ChunkSize: Specifies the maximum size requested, in bytes, of documents returned by this stored procedure. If the document size is larger than this maximum size, only the first **@ChunkSize** bytes MUST be returned for **{Content}** and the front-end Web server can request the remainder of the document in a subsequent operation.

@PageSize: The maximum number of records to be returned from a single call to this stored procedure.

@FromSiteId: The site collection identifier of the document stream where record retrieval begins.

@FromDeleteTransactionId: The delete transaction identifier of the document stream where record retrieval begins.

@FromParentId: The document identifier (2) of the parent container of the document for the document stream where record retrieval begins.

@FromId: The document identifier (2) for the document stream where record retrieval begins.

@FromLevel: The publishing level for the document stream where record retrieval begins.

Return values: An integer that MUST be zero.

Result sets: MUST return the **Document Stream Storage Migration Result Set** (section [2.2.6.5](#)), with one row for each document stream in ascending order by (**SiteId**, **DeleteTransactionId**, **ParentId**, **Id**, **Level**) up to the specified **@PageSize** number of rows. Document streams MUST only be returned for documents not in external storage.

3.1.4.117 `proc_GetDocStreamsForInlineMigration`

The `proc_GetDocStreamsForInlineMigration` stored procedure is called to migrate document streams to storage in the content database. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_GetDocStreamsForInlineMigration (

```

```

@ChunkSize          int,
@PageSize           int,
@FromSiteId        uniqueidentifier,
@FromDeleteTransactionId varbinary(16),
@FromParentId      uniqueidentifier,
@FromId            uniqueidentifier,
@FromLevel         tinyint
);

```

@ChunkSize: Specifies the maximum size requested, in bytes, of documents returned by this stored procedure. If the document size is larger than this maximum size, only the first **@ChunkSize** bytes MUST be returned for **{Content}** and the front-end Web server can request the remainder of the document in a subsequent operation.

@PageSize: The maximum number of records to be returned from a single call to this stored procedure.

@FromSiteId: The site collection identifier of the document stream where record retrieval begins.

@FromDeleteTransactionId: The delete transaction identifier of the document stream where record retrieval begins.

@FromParentId: The document identifier (2) of the parent container of the document for the document stream where record retrieval begins.

@FromId: The document identifier (2) for the document stream where record retrieval begins.

@FromLevel: The publishing level for the document stream where record retrieval begins.

Return values: An integer that MUST be zero.

Result sets: MUST return **the Document Stream Storage Migration Result Set** (section [2.2.6.5](#)), with one row for each document stream in ascending order by (**SiteId**, **DeleteTransactionId**, **ParentId**, **Id**, **Level**) up to the specified **@PageSize** number of rows. Document streams MUST only be returned for documents in remote BLOB storage or external storage.

3.1.4.118 proc_GetDocStreamsForRbsMigration

The **proc_GetDocStreamsForRbsMigration** stored procedure is called to migrate document streams to remote BLOB storage. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_GetDocStreamsForRbsMigration (
@DefaultRbsProviderName sysname,
@ChunkSize          int,
@PageSize           int,
@FromSiteId        uniqueidentifier,
@FromDeleteTransactionId varbinary(16),
@FromParentId      uniqueidentifier,
@FromId            uniqueidentifier,
@FromLevel         tinyint
);

```

@DefaultRbsProviderName: Specifies the remote BLOB storage store name for which records MUST be excluded from the result set.

@ChunkSize: Specifies the maximum size requested, in bytes, of documents returned by this stored procedure. If the document size is larger than this maximum size, only the first **@ChunkSize** bytes MUST be returned for **{Content}** and the front-end Web server can request the remainder of the document in a subsequent operation.

@PageSize: The maximum number of records to be returned from a single call to this stored procedure.

@FromSiteId: The site collection identifier of the document stream where record retrieval begins.

@FromDeleteTransactionId: The delete transaction identifier of the document stream where record retrieval begins.

@FromParentId: The document identifier (2) of the parent container of the document for the document stream where record retrieval begins.

@FromId: The document identifier (2) for the document stream where record retrieval begins.

@FromLevel: The publishing level for the document stream where record retrieval begins.

Return values: An integer that MUST be zero.

Result sets: MUST return the **Remote Blob Storage Document Stream Migration Result Set** (section [3.1.4.118.1](#)), with one row for each document stream in ascending order by (**SiteId, DeleteTransactionId, ParentId, Id, Level**), for rows where this key is strictly larger than (**@FromSiteId, @FromDeleteTransactionId, @FromParentId, @FromId, @FromLevel**), with component-wise comparison, up to the specified **@PageSize** number of rows. Document streams MUST only be returned for documents whose remote BLOB storage store name does not match **@DefaultRbsProviderName** or those not in remote BLOB storage.

3.1.4.118.1 Remote Blob Storage Document Stream Migration Result Set

The **Remote Blob Storage Document Stream Migration Result Set** returns information about the document streams in a site collection for remote data store management. Each row corresponds to a document stream along with associated metadata. The T-SQL syntax for the result set is as follows.

Size	int,
DocFlags	int,
{Content}	varbinary(max),
{RbsResReference}	varbinary(800),
Id	uniqueidentifier,
InternalVersion	int,
Level	tinyint,
SiteId	uniqueidentifier,
DeleteTransactionId	uniqueidentifier,
ParentId	uniqueidentifier,
RbsCollectionId	int

Size: The size, in bytes, of the document stream of the document.

DocFlags: The [Document Flags](#) for the document.

{Content}: The content stream of the document. For an uncustomized document, or if the document has external storage, this MUST be NULL. Otherwise, if the content is larger than the

value specified in the *@ChunkSize* parameter, only the first *@ChunkSize* bytes MUST be returned, and the front-end Web server can request individual chunks of content in a subsequent request.

{RbsResReference}: If remote BLOB storage is enabled and the content of the document is contained in a remote data store, this MUST be the remote BLOB storage identifier for the content of the document. If remote BLOB storage is disabled or the content of the document is not contained in a remote data store, this MUST be NULL. For more information about remote BLOB storage, see [\[MS-WSSO\]](#) section 2.1.2.3.8.

Id: The document identifier (2) of the document.

InternalVersion: The internal version number of the document.

Level: The publishing level of the document.

SiteId: The site collection identifier of the site collection that contains the document.

DeleteTransactionId: The delete transaction identifier of the document.

ParentId: The document identifier (2) of the parent container of the document.

RbsCollectionId: The identifier for the remote BLOB storage collection of site collection that contains the document.

3.1.4.119 **proc_GetDocVersionStreamsForExternalMigration**

The **proc_GetDocVersionStreamsForInlineMigration** stored procedure is called to migrate document streams for document versions to external storage. The T-SQL syntax for the stored procedure is as follows.

```
PROCEDURE proc_GetDocVersionStreamsForExternalMigration (  
    @ChunkSize          int,  
    @PageSize           int,  
    @FromSiteId         uniqueidentifier,  
    @FromId             uniqueidentifier,  
    @FromUIVersion      int  
);
```

@ChunkSize: Specifies the maximum size requested, in bytes, of documents returned by this stored procedure. If the document size is larger than this maximum size, only the first **@ChunkSize** bytes MUST be returned for **{Content}** and the front-end Web server can request the remainder of the document in a subsequent operation.

@PageSize: The maximum number of records to be returned from a single call to this stored procedure.

@FromSiteId: The site collection identifier of the document stream where record retrieval begins.

@FromId: The document identifier (2) for the document stream of the document version where record retrieval begins.

@FromUIVersion: The UI version number where record retrieval begins.

Return values: An integer that MUST be zero.

Result sets: MUST return the **Document Version Stream Storage Migration Result Set** (section [2.2.6.6](#)), with one row for each document stream in ascending order by (**SiteId**, **Id**,

UIVersion) up to the specified **@PageSize** number of rows. Document streams MUST only be returned for documents not in external storage.

3.1.4.120 **proc_GetDocVersionStreamsForInlineMigration**

The **proc_GetDocVersionStreamsForInlineMigration** stored procedure is called to migrate document streams for document versions to storage in the content database. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetDocVersionStreamsForInlineMigration (  
    @ChunkSize          int,  
    @PageSize           int,  
    @FromSiteId        uniqueidentifier,  
    @FromId             uniqueidentifier,  
    @FromUIVersion     int  
);
```

@ChunkSize: Specifies the maximum size requested, in bytes, of documents returned by this stored procedure. If the document size is larger than this maximum size, only the first **@ChunkSize** bytes MUST be returned for **{Content}** and the front-end Web server can request the remainder of the document in a subsequent operation.

@PageSize: The maximum number of records to be returned from a single call to this stored procedure.

@FromSiteId: The site collection identifier of the document stream where record retrieval begins.

@FromId: The document identifier (2) for the document stream of the document version where record retrieval begins.

@FromUIVersion: The UI version number where record retrieval begins.

Return values: An integer that MUST be zero.

Result sets: MUST return the **Document Version Stream Storage Migration Result Set** (section [2.2.6.6](#)), with one row for each document stream in ascending order by (**SiteId, Id, UIVersion**) up to the specified **@PageSize** number of rows. Document streams MUST only be returned for documents in external storage or remote BLOB storage.

3.1.4.121 **proc_GetDocVersionStreamsForRbsMigration**

The **proc_GetDocVersionStreamsForRbsMigration** stored procedure is called to migrate document streams for document versions to remote BLOB storage. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetDocVersionStreamsForRbsMigration (  
    @DefaultRbsProviderName sysname,  
    @ChunkSize              int,  
    @PageSize               int,  
    @FromSiteId            uniqueidentifier,  
    @FromId                 uniqueidentifier,  
    @FromUIVersion         int  
);
```

@DefaultRbsProviderName: Specifies the remote BLOB storage store name for which records MUST be excluded from the result set.

@ChunkSize: Specifies the maximum size requested, in bytes, of documents returned by this stored procedure. If the document size is larger than this maximum size, only the first **@ChunkSize** bytes a MUST be returned for **{Content}** and the front-end Web server can request the remainder of the document in a subsequent operation.

@PageSize: The maximum number of records to be returned from a single call to this stored procedure.

@FromSiteId: The site collection identifier of the document stream where record retrieval begins.

@FromId: The document identifier (2) for the document stream of the document version where record retrieval begins.

@FromUIVersion: The UI version number where record retrieval begins.

Return values: An integer that MUST be zero.

Result sets: MUST return the **Remote Blob Storage Document Version Stream Migration Result Set** (section [3.1.4.121.1](#)), with one row for each document stream in ascending order by (**SiteId, Id, UIVersion**). Document streams MUST only be returned for documents whose remote BLOB storage store name does not match **@DefaultRbsProviderName** or that are not in remote BLOB storage.

3.1.4.121.1 Remote Blob Storage Document Version Stream Migration Result Set

The **Remote Blob Storage Document Version Stream Migration Result Set** returns information about the document streams for document versions in a site collection for remote data store management. Each row corresponds to a document stream for a document version along with associated metadata. The T-SQL syntax for the result set is as follows.

Size	int,
DocFlags	int,
{Content}	varbinary(max),
{RbsResReference}	varbinary(800),
Id	uniqueidentifier,
InternalVersion	int,
UIVersion	int,
SiteId	uniqueidentifier,
RbsCollectionId	int

Size: The size, in bytes, of the document stream of the document version.

DocFlags: The [Document Flags](#) for the document version.

{Content}: The content stream of the document version. For an uncustomized document or if the document has external storage, this MUST be NULL. Otherwise, if the content is larger than the value specified in the **@ChunkSize** parameter, only the first **@ChunkSize** bytes MUST be returned, and the front-end Web server can request individual chunks of content in a subsequent request.

{RbsResReference}: If remote BLOB storage is enabled and the content of the document is contained in a remote data store, this MUST be the remote BLOB storage identifier for the document's content. If remote BLOB storage is disabled or the content of the document is not

contained in a remote data store, this MUST be NULL. For more information about remote BLOB storage, see [\[MS-WSSO\]](#) section 2.1.2.3.8.

Id: The document identifier (2) of the document version.

InternalVersion: The internal version number of the document version.

UIVersion: The UI version number of the document version.

SiteId: The site collection identifier of the site collection that contains the document.

RbsCollectionId: The identifier for the remote BLOB storage collection of site collection that contains the document.

3.1.4.122 `proc_GetStreamsForBackup`

The `proc_GetStreamsForBackup` stored procedure is called to retrieve all document streams for a site collection for the purposes of site collection backup. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetStreamsForBackup (  
    @SiteId                uniqueidentifier,  
    @ChunkSize             int  
);
```

@SiteId: The site collection identifier for which document streams are retrieved.

@ChunkSize: Specifies the maximum size requested, in bytes, of documents returned by this stored procedure. If the document size is larger than this maximum size, only the first **@ChunkSize** bytes MUST be returned for **{Content}** and the front-end Web server can request the remainder of the document in a subsequent operation.

Return values: An integer that MUST be zero.

Result sets: MUST return the following result sets.

3.1.4.122.1 Document Content Stream Backup Result Set

The **Document Content Stream Backup Result Set** MUST only contain rows for documents with non-empty document streams; see [Document Content Stream Transfer Result Set](#) (section 2.2.6.3).

3.1.4.122.2 Document Version Content Stream Backup Result Set

The **Document Version Content Stream Backup Result Set** MUST only contain rows for documents with non-empty document streams; see [Document Version Content Stream Transfer Result Set](#) (section 2.2.6.4).

3.1.4.123 `proc_GetStreamsForSiteMove`

The `proc_GetStreamsForSiteMove` stored procedure is called to retrieve metadata about all document streams stored in remote BLOB storage for a site collection for the purposes of moving site collections between content databases. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetStreamsForSiteMove (  
    @SiteId                uniqueidentifier,
```



```
    @ChunkSize          int
);
```

@SiteId: The site collection identifier for which document streams are retrieved.

@ChunkSize: Specifies the maximum size requested, in bytes, of documents returned by this stored procedure. If the document size is larger than this maximum size, only the first **@ChunkSize** bytes MUST be returned for **{Content}** and the front-end Web server can request the remainder of the document in a subsequent operation.

Return values: An integer that MUST be zero.

Result sets: MUST return the following result sets.

3.1.4.123.1 Document Content Stream Site Move Result Set

The **Document Content Stream Site Move Result Set** MUST only contain rows for documents in remote BLOB storage; see [Document Content Stream Transfer Result Set](#) (section 2.2.6.3).

3.1.4.123.2 Document Version Content Stream Site Move Result Set

The **Document Version Content Stream Site Move Result Set** MUST only contain rows for documents in remote BLOB storage; see [Document Version Content Stream Transfer Result Set](#) (section 2.2.6.4).

3.1.4.124 proc_EnsureSiteRbsCollection

The **proc_EnsureSiteRbsCollection** stored procedure is called to ensure that a site collection has a remote BLOB storage collection. The T-SQL syntax for the stored procedure is as follows.

```
PROCEDURE proc_EnsureSiteRbsCollection (
    @SiteId uniqueidentifier,
    @Force bit,
    @RbsCollectionId int OUTPUT,
    @RequestGuid uniqueidentifier = null OUTPUT
);
```

@SiteId: The site collection identifier for the site collection on which the stored procedure operates.

@Force: A bit that controls the behavior of the stored procedure if the site collection already has a remote BLOB storage collection allocated. If the value is zero, it MUST NOT allocate a new remote BLOB storage collection; otherwise it MUST.

@RbsCollectionId: The identifier for the remote BLOB storage collection of the site collection.

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

Return values: An integer that MUST be zero.

Result sets: MUST NOT return a result set.

3.1.4.125 proc_GetStreamsForPreSiteMove

The **proc_GetStreamsForPreSiteMove** stored procedure is called to retrieve metadata about all document streams stored in remote BLOB storage for a site collection to move site collections between content databases. The T-SQL syntax for the stored procedure is as follows.

```
PROCEDURE proc_GetStreamsForSiteMove (
    @SiteId                uniqueidentifier,
    @RbsStoreList          varbinary(MAX)
);
```

@SiteId: The site collection identifier for which document streams are retrieved.

@RbsStoreList: A packed array of 32-bit big-endian integers. Each integer MUST be the identifier in the source database for a remote BLOB storage store that exists in both databases in the move operation.

Return values: An integer that MUST be zero.

Result sets: MUST return the following result sets.

3.1.4.125.1 Remote Blob Collection Identifier Preliminary Site Move Result Set

The **Remote Blob Storage Collection Identifier Preliminary Site Move Result Set** returns information about the remote BLOB storage collection for the site collection specified by the @SiteId parameter. It contains a single row. The T-SQL syntax for the result set is as follows:

```
RbsCollectionId    int
```

RbsCollectionId: The identifier for the remote BLOB storage collection of the site collection specified by the @SiteId parameter.

3.1.4.125.2 Document Content Preliminary Site Move Result Set

The **Document Content Preliminary Site Move Result Set** MUST only contain rows for documents in remote BLOB storage and MUST NOT both belong to the remote BLOB storage collection of the site collection specified by the @SiteId parameter and be stored in one of the remote BLOB storage stores specified by the @RbsStoreList parameter; see section [2.2.6.3](#).

3.1.4.125.3 Document Version Content Preliminary Site Move Result Set

The **Document Version Content Preliminary Site Move Result Set** MUST only contain rows for documents in remote BLOB storage and MUST NOT both belong to the remote BLOB storage collection of the site collection specified by the @SiteId parameter and be stored in one of the remote BLOB storage stores specified by the @RbsStoreList parameter; see section [2.2.6.4](#).

3.1.4.126 proc_ListRbsStoresWithIds

The **proc_ListRbsStoresWithIds** stored procedure enumerates the remote BLOB storage stores with which the back-end database server has been configured, if any. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_GetStreamsForBackup (
    @RequestGuid          uniqueidentifier = NULL OUTPUT
);

```

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

Return values: An integer that MUST be zero.

Result sets: MUST return the following result sets.

3.1.4.126.1 List Remote Blob Storage Stores With Identifiers Result Set

The **List Remote Blob Storage Stores With Identifiers Result Set** lists the names and the identifiers for the remote BLOB storage stores. The T-SQL syntax for the result set is as follows:

```

blob_store_name          sysname,
blob_store_id            smallint;

```

blob_store_name: The name of a remote BLOB storage store.

blob_store_id: The identifier of a remote BLOB storage store.

3.1.4.127 proc_ShallowCopyRbsBlobs

The **proc_ShallowCopyRbsBlobs** stored procedure transfers registration of BLOBs stored in remote BLOB storage from one back-end database server to another. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE dbo.proc_ShallowCopyRbsBlobs
(
    @SiteId uniqueidentifier,
    @RbsSourceStoreList varbinary(max),
    @RbsTargetStoreList varbinary(max),
    @TargetDbName sysname
);

```

@SiteId: The site collection identifier of the site collection that is being copied.

@RbsSourceStoreList: A packed array of 32-bit big-endian integers. Each integer MUST be the identifier in the source database for a remote BLOB storage store that exists in both databases in the move operation. The entries in the array MUST correspond to the entries specified in the **@RbsTargetStoreList** parameter in the same order.

@RbsTargetStoreList: A packed array of 32-bit big-endian integers. Each integer MUST be the identifier in the destination database for a remote BLOB storage store that exists in both databases in the move operation. The entries in the array MUST correspond to the entries specified in the **@RbsTargetStoreList** parameter in the same order.

@TargetDbName: The name of the destination database in the move operation.

Return values: An integer that MUST be zero.

Result sets: MUST NOT return any result sets.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

3.2 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

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 front-end Web server can maintain the following sets of data for this protocol within object structures. There is no requirement for the state within these structures to be a complete representation of all data maintained on the back-end database server, but can be populated as various requests to the back-end database server are fulfilled. Data maintained on the front-end Web server can be discarded after individual sequences of requests have finished as part of a response for a higher level event.

- Configuration
- Site Collections
- Sites
- Lists
- List Items
- Documents
- Users
- Groups

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 following:

- As part of link fixup of a list item, the protocol client updates the **AllUserData** table define in [\[MS-WSSFO2\]](#), section [2.2.7.3](#), to store the updated field data.
- When updating the values for **calculated fields** on a list item, the protocol client directly updates the sql_variant column of **AllUserData** where the calculation results are stored.
- When enabling an indexed field on a list, the protocol client directly updates the **NameValuePair** table (or one of the collated **NameValuePair** tables for textual fields, based on the collation order of the site that contains the list), copying the indexed field data from the **AllUserData** table for all items in the list. When disabling an indexed field on a list, the protocol client deletes all rows from the table.
- As part of updating item order in an ordered list, the protocol client updates the tp_ItemOrder column of the **AllUserData** table to 1.79E plus 308 for all items being reordered before invoking **proc_UpdateOrderNumber** on each of those list items.

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

This section provides specific example scenarios. These examples describe in detail the process of communication between the various server features involved in the Windows® SharePoint® Services deployment. In conjunction with the detailed protocol documentation described in the reference documents, this information is intended to provide a comprehensive view of how Windows SharePoint Services front-end Web servers communicate with both EUC and back-end database server systems.

4.1 Change Log

The stored procedures in this protocol example do not require special sequencing when they are called. The example illustrated in the following shows an application calls **proc_GetChanges** to get a list of events and their metadata.

User uploads a document to a document library. This effectively adds a list item to the document library, which should cause an event to be appended to the change log.

The application calls **proc_GetChanges** with the following parameter, with the intention to retrieve any events that involves an item being added and occurs between Coordinated Universal Time (UTC) 2008/02/07 and UTC 2008/02/08 on a particular list.

- **Site Collection Id:** '61854258-1D17-410E-8363-ADC6C0B5C6D4'
- **Site Id:** '2FF0E4EC-B41B-412E-AEDF-C796BBF0D905'
- **List Id:** '27AC1BC8-BAF5-418A-8634-F31A9A8886D5'
- **Start Time Stamp:** '2008-02-07'
- **Start Change Log Id:** NULL
- **End Time Stamp:** '2008-02-08'
- **End Change Log Id:** NULL
- **Event Object Type Flags:** 1 (means list item)

Event Type Flags: 4096 (Add)

The **EventInformation Result Set** (section [3.1.4.51.1](#)) is returned which contains the time stamp and the change log identifier of the earliest event, similar to the following:

- The time stamp of the first event in the change log is UTC 2008-02-06 22:10:08.460

The identifier of the first event in the change log is 1.

The **EventDetails Result Set** (section [3.1.4.51.2](#)) is returned, which contains 1 event and its metadata

- **EventTime:** 2008-02-07 19:06:48.943
- **Id:** 2159
- **SiteId:** 61854258-1D17-410E-8363-ADC6C0B5C6D4
- **WebId:** 2FF0E4EC-B41B-412E-AEDF-C796BBF0D905

- **ListId:** 27AC1BC8-BAF5-418A-8634-F31A9A8886D5
- **ItemId:** 1
- **DocId:** 3705DD61-8DB6-4C7B-AF2B-571E45721F8C
- **Guid0:** NULL
- **Int0:** NULL
- **ContentTypeId:** NULL
- **ItemFullUrl:** Shared Documents/myfile.doc
- **EventType:** 4097
- **ObjectType:** 1
- **TimeLastModified:** 2008-02-07 19:06:47.000
- **Int1:** NULL

This result set shows that during the time period requested by the application, one event in the change log matches the search criteria. It is a document named "myfile.doc", which was added to this document library around 2008-02-07 19:06:48.943.

4.2 Link Fixup

In the following example, a single item is dirty in a list with identifier of 43E3226F-55A6-41BC-A194-9DD74AF4A1D5 in a site with an identifier of 2EF8C46F-D91F-46C1-8D9B-9A24A62AA268 whose root folder is Lists/Links.

The protocol client sends:

```
SET NOCOUNT ON;EXEC proc_StartUndirtyList '2EF8C46F-D91F-46C1-8D9B-9A24A62AA268','43E3226F-55A6-41BC-A194-9DD74AF4A1D5','B470CF0A-150F-41BC-9329-68BB7D427DA6';SELECT TOP 1000 U.nvarchar3,U.tp_DirName,U.tp_LeafName,U.tp_Level FROM UserData AS U INNER JOIN Docs AS D ON U.tp_SiteId=D.SiteId AND U.tp_DirName=D.DirName AND U.tp_LeafName=D.LeafName AND U.tp_Level=D.Level AND D.ListDataDirty = 1 WHERE U.tp_SiteId='4AB8AC94-9F0A-44D3-B7DC-A81A323A6BDF' AND (U.tp_DirName=N'Lists/Links' OR U.tp_DirName LIKE N'Lists/Links/%') AND U.tp_RowOrdinal=0 ORDER BY U.tp_DirName Asc, U.tp_LeafName Asc, U.tp_Level
```

To which the protocol server returns the following row set:

- **nvarchar3:** /Shared%20Documents/old%20test.txt
- **tp_DirName:** Lists/Links
- **tp_LeafName:** 1_.000
- **tp_Level:** 1

Indicating that a list item with a (**DirName, LeafName, Level**) key of (List/Links, 1_.000, 1) has invalid field data in **nvarchar3** that is /Shared%20Documents/old%20test.txt. Because fewer than 1000 rows were returned, the protocol client determines that it need not continue to perform link fixup on the list.

The protocol client then calls:

```
EXEC proc_GetListDataLinks '4AB8AC94-9F0A-44D3-B7DC-A81A323A6BDF','2EF8C46F-D91F-46C1-8D9B-9A24A62AA268',N'Lists/Links',N'1_.000',1,N'Lists/Links',N'1_.000',1,1
```

Causing the protocol server to respond with an empty **Web List For Normalization Result Set** (section [3.1.4.58.1](#)) and the following **List Data Link Information Result Set** (section [3.1.4.58.2](#)):

- **DirName:** Lists/Links
- **LeafName:** 1_.000
- **Level:** 1
- **FieldId:** C29E077D-F466-4D8E-8BBE-72B66C5F205C
- **TargetDirName:** Shared Documents
- **TargetLeafName:** test.txt
- **Type:** 74
- **Security:** 85
- **Dynamic:** 83
- **ServerRel:** 1
- **Type:** 0
- **PointsToDir:** 0

The first result set indicates that the site has no subsites. The second result set specifies that the new link data should be Shared Documents/text.txt

This allows the protocol client to compute the correct field data and marked not dirty the list item via the following request:

```
exec sp_executesql N'DECLARE @@iRet int;SET @@iRet=0;BEGIN TRAN;UPDATE UserData SET nvarchar3=@P1 WHERE tp_SiteId='4AB8AC94-9F0A-44D3-B7DC-A81A323A6BDF' AND tp_DirName=N'Lists/Links' AND tp_LeafName=N'1_.000' AND tp_Level=1 AND tp_RowOrdinal=0;EXEC @@iRet = proc_UndirtyListItem '4AB8AC94-9F0A-44D3-B7DC-A81A323A6BDF',N'Lists/Links',N'1_.000',1,'B470CF0A-150F-41BC-9329-68BB7D427DA6',NULL; IF @@iRet <> 0 GOTO done;done:IF @@iRet <> 0 ROLLBACK TRAN;ELSE BEGIN COMMIT TRAN;EXEC proc_FinishUndirtyList '2EF8C46F-D91F-46C1-8D9B-9A24A62AA268','43E3226F-55A6-41BC-A194-9DD74AF4A1D5','B470CF0A-150F-41BC-9329-68BB7D427DA6',1;END', N'@P1 nvarchar(28)', N'/Shared%20Documents/test.txt'
```

which generates no row sets.

4.3 Themes

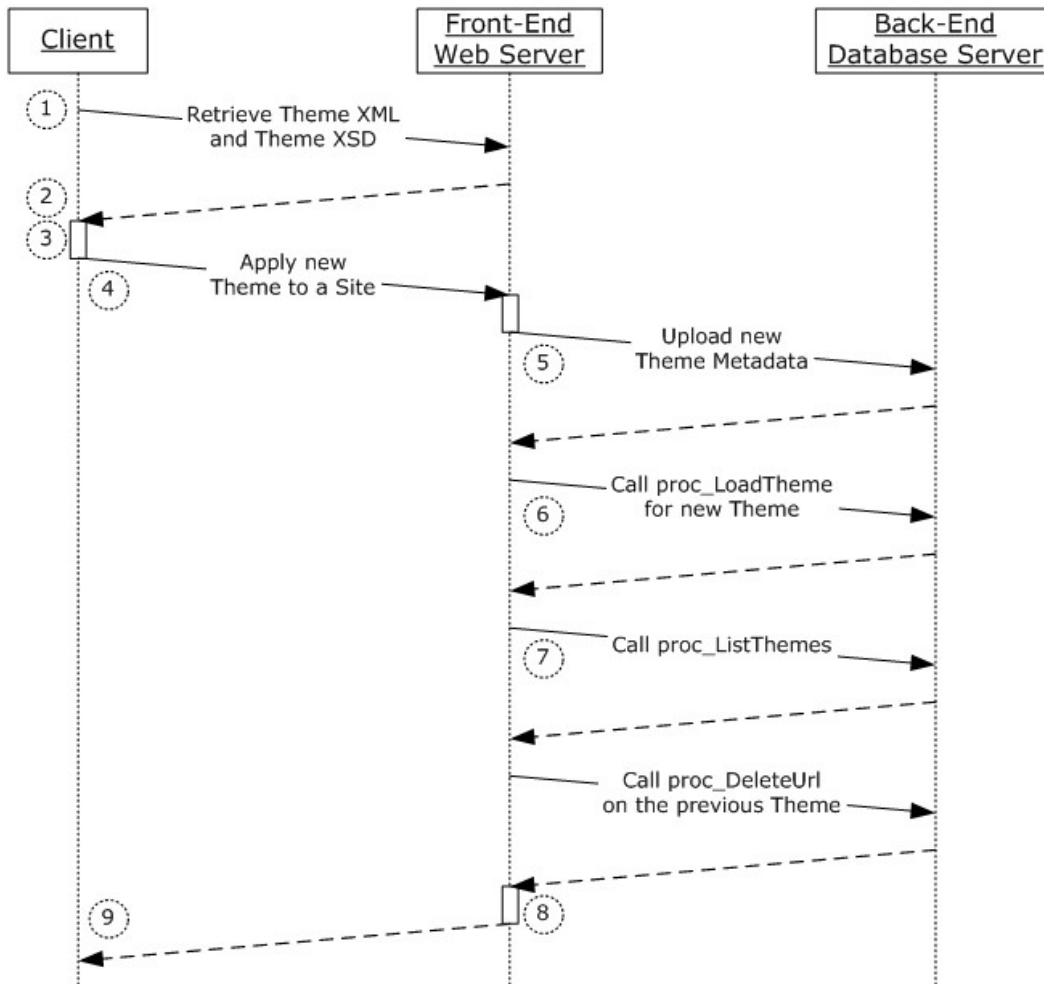


Figure 7: Themes

This diagram illustrates the process of applying a new theme to a site. The following is a more in-depth explanation of this messaging sequence process:

1. The client's Web browser retrieves the theme XML file and theme XSD file from the front-end Web server.
2. The client's Web browser displays the list of theme names from the theme XML file to the user.
3. Using the client's Web browser, the end user selects a theme name to be applied to the site.
4. The client's Web browser requests the front-end Web server to apply the new theme to the site using the theme name.
5. For each theme file found at the theme installation path on the front-end Web server, the front-end Web server calls the [proc_AddGhostDocument](#) stored procedure so that all theme metadata is uploaded to the content database on the back-end database server.

6. The front-end Web server then calls the [proc_LoadTheme](#) stored procedure (section [3.1.4.76](#)) to retrieve the **Theme Files Information Result Set** (section [3.1.4.76.1](#)) and the **Theme INF File Information Result Set** (section [3.1.4.76.2](#)).
7. The front-end Web server then calls the [proc_ListThemes](#) (section [3.1.4.75](#)) and the **proc_DeleteUrl** (defined in [\[MS-WSSFO2\]](#) section 3.1.5.13) stored procedures to remove the previous theme from the site (2). For each row returned the by **Theme Information Result Set** (section [3.1.4.75.1](#)) returned by the `proc_ListThemes` stored procedure, the **proc_DeleteUrl** stored procedure is called to remove the previous theme from the site (2). The **proc_DeleteUrl** stored procedure is not called for the theme just applied to the site (2).
8. The front-end Web server then applies the theme data returned by the **Theme Files Information Result Set** and the **Theme INF File Information Result Set** from the `proc_LoadTheme` stored procedure to the current page requested by the client's Web browser.
9. The front-end Web server returns the current page to the client's Web browser.

4.4 Add Just Enough Fields to Cause Allocation of Additional Rows Per List Item

This scenario is initiated when a list or a wide list already contains the maximum number of fields for a type per row in the content database, and a field of that type is added to the list or wide list.

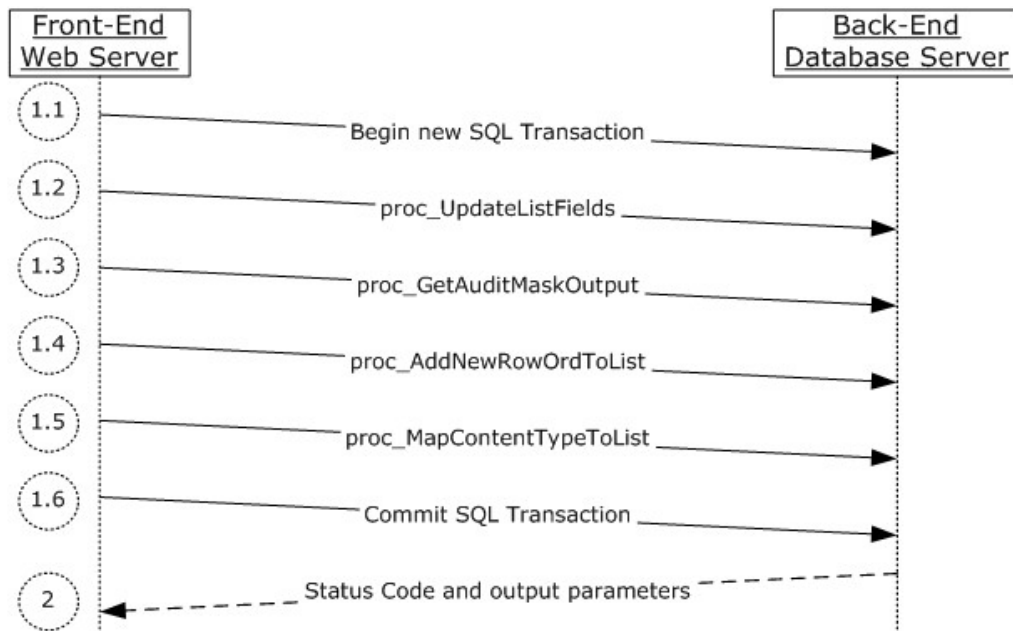


Figure 8: Adding just enough fields to cause allocation of additional rows per list item

The following actions happen:

1. The front-end Web server builds a transactional dynamic query in T-SQL syntax to add the field to the list or wide list and add an additional row per list item in the content database. This query is sent using TDS.
 1. The query begins a new transaction.

2. The query attempts to add the field to the list or wide list using the **proc_UpdateListFields** stored procedure described in [\[MS-WSSCCSP\]](#), section 3.1.5.64.
 3. The query gets the Audit Mask Information for the list or wide list to which the field is being added using the **proc_GetAuditMaskOutput** stored procedure described in [\[MS-WSSFO2\]](#) section 3.1.5.26.
 4. The query attempts to allocate an additional row per list item, specifying an incremented row ordinal count for the row using the **proc_AddNewRowOrdToList** stored procedure.
 5. The query records that the newly added field is in use in the list or wide list is using the **proc_MapContentTypeToList** stored procedure described in [\[MS-WSSCCSP2\]](#) section 3.1.4.48.
 6. The query rolls back the transaction if the previous actions were not successful, or it commits the transaction if they were successful.
2. The back-end database server returns one result set which contains the return code of the actions in the query and the output parameters from the **proc_GetAuditMaskOutput** stored procedure.

4.5 Allocate New Rows While Inserting an Item into a Wide List

This can happen if a field that previously caused an additional row to be allocated per list item was deleted, and a list item is added to the wide list. This scenario is initiated if a row needs to be allocated for a list item in the content database if a row of a specific row ordinal in a wide list does not correspond to any of the fields in the wide list.

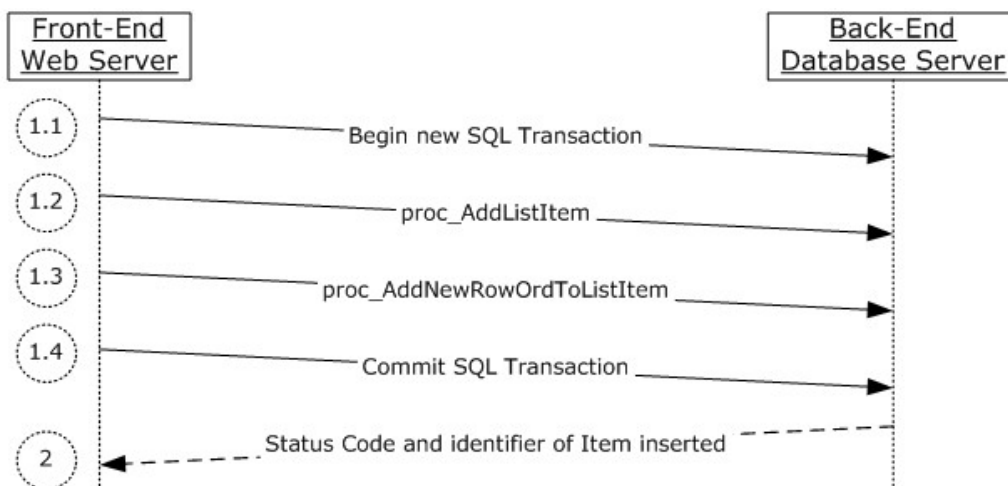


Figure 9: Allocating new rows for a list item while inserting a list

For simplicity's sake, this example assumes:

- The wide list does not contain any indexed fields, lookup fields, and calculated fields.
- The wide list does not have any alerts associated with it.
- The wide list is not a list of meeting attendees.
- The list item being added does not require any delayed link fixup.

The following actions happen:

1. The front-end Web server builds a transactional dynamic query in SQL syntax to add a List Item and any rows not corresponding to any fields to a wide list. This query is sent using TDS.
 1. The query begins a new transaction.
 2. The query attempts to add a row for each row ordinal that has at least one field corresponding to the row using the **proc_AddListItem** stored procedure described in [\[MS-WSSFO2\]](#) section 3.1.5.4.
 3. The query attempts to add a row for each row ordinal that does not have even a single Field corresponding to it using the **proc_AddNewRowOrdToListItem** stored procedure.
 4. The query rolls back the transaction if the previous actions were not successful, or it commits the transaction if they were successful.
2. The back-end database server returns one result set which contains the return code of the actions in the query and the identifier of the list item being added.

4.6 Cascade Delete Relationship Behavior

This is used when deleting an item with a lookup field and cascade deleting all related target list items based on that field. This scenario is used when a list containing the item to be deleted contains a column where restrict delete relationship behavior is enforced.

This example assumes:

- There is one list (child list) with a lookup field that points to a target list (parent list).
- The parent list contains the parent items and the child list contains the child items.
- There are one or more child items looking up to a parent item in the parent list.

The following actions occur:

- The front-end Web server builds a transactional dynamic query in SQL syntax to delete a parent list item and have all related child items deleted as well.
 1. The site is locked using **proc-TranLockWeb**. This prevents modifications in any other lists in the same site with relationship lookup fields with cascade or restrict relationship behaviors while the cascade delete operation is in progress while the transaction is in progress.
 2. The query begins a new SQL transaction.
 3. To get a list of all cascade child items that are to be deleted, **proc_GetListItemsTreeToDelete** is called.
 4. To prevent extra round trips to the SQL database, **proc_EnumListsWithMetadata** (as specified in [\[MS-WSSCCSP2\]](#) section 3.1.4.9) is called on the child lists to pre-cache them in memory.
 5. The items are then deleted with the **proc_CascadeDeleteItems** call. This will delete all the related items in the child list (child items) and the parent item.
 6. The lists in the site are then unlocked using **proc-TranUnlockWeb**.

7. The query rolls back the transaction if the previous actions were not successful, or it commits the transaction if they were successful.

4.7 File Fragment Update Behavior

This is used to grab file fragments, modify, and then re-apply the changes back to the original document.

The following actions happen:

- The front-end Web server calls a sequence of stored procedures to get, modify, and close a file fragment:
 1. First, **proc_GetDocsMetaInfo** ([\[MS-WSSFO2\]](#) section 3.1.5.30) is called to load the document metadata into front-end Web server memory.
 2. Next, **proc_GetFileFragmentsByTag** is called to get a fragment up to 8192 bytes in size. This fragment can then be modified by the client. This will also get any updates to the specified fragment by another user, as described in the use of **proc_SaveFileFragmentById**.
 3. When all modifications have finished and the client is ready to submit changes, **proc_FetchDocForHttpGet** ([\[MS-WSSFO2\]](#) section 3.1.5.19) is first called to fetch the complete latest document. The client then takes the fragments it has and re-assembles the document.
 4. Once the document is assembled and ready to save, **proc_SaveFileFragmentById** is called to save all fragment changes to the file fragments table. This does not mean it is re-integrated into the document. This is just used so other users who call **proc_GetFileFragmentsByTag** can get the changes made by the user.
 5. Once all fragments are saved, **proc_FetchDocForUpdate** ([\[MS-WSSFO2\]](#) section 3.1.5.21) is then called to get the document for update.
 6. Finally, **proc_UpdateDocument** ([\[MS-WSSFO2\]](#) section 3.1.5.128) is called to save the file and all changes back to the original document in the document library.

5 Security

5.1 Security Considerations for Implementers

Security for this protocol is controlled by the permissions to the databases on the back-end database server, which is negotiated as part of the Tabular Data Stream [\[MS-TDS\]](#) protocol.

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 permissions, 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

None.

6 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- Microsoft® SharePoint® Foundation 2010
- Windows® SharePoint® Services 3.0
- Windows® SharePoint® Services 2.0
- Windows® SharePoint® Services
- Microsoft® SQL Server® 2005
- Microsoft® SQL Server® 2008
- Microsoft® SQL Server® 2008 R2

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

7 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

8 Index

A

Abstract data model
[client](#) 228
[server](#) 53
[Add just enough fields to cause allocation of additional rows per list item example](#) 234
[Allocate new rows while inserting an item into a wide list example](#) 235
[AllUserData table structure](#) 50
[Applicability](#) 17
[Attribute groups - overview](#) 52
[Attributes - overview](#) 52

B

[Binary structures - overview](#) 46

C

[Capability negotiation](#) 17
[Cascade delete relationship behavior example](#) 236
[Change log ContentTypeId simple type](#) 30
[Change log DocId simple type](#) 23
[Change log example](#) 230
[Change log Guid0 simple type](#) 24
[Change log Int0 simple type](#) 27
[Change log Int1 simple type](#) 38
[Change log ItemFullUrl simple type](#) 31
[Change log ItemId simple type](#) 21
[Change log ItemName simple type](#) 36
[Change log ListId simple type](#) 18
[Change log operations](#) 16
[server](#) 53
[Change log SiteId simple type](#) 41
[Change log TimeLastModified simple type](#) 33
[Change log WebId simple type](#) 42
[Change tracking](#) 240
[Check-in and check-out operations](#) 16
[server](#) 53

Client

[abstract data model](#) 228
[initialization](#) 228
[local events](#) 229
[message processing](#) 229
[overview](#) 228
[sequencing rules](#) 229
[timer events](#) 229
[timers](#) 228
[WSSDLIM interface](#) 228
[Collated NameValuePair tables](#) 51
Common data types
[overview](#) 18
Complex types
[List Data Source](#) 52

D

Data model - abstract

[client](#) 228
[server](#) 53

Data types

[change log ContentTypeId simple type](#) 30
[change log DocId simple type](#) 23
[change log Guid0 simple type](#) 24
[change log Int0 simple type](#) 27
[change log Int1 simple type](#) 38
[change log ItemFullUrl simple type](#) 31
[change log ItemId simple type](#) 21
[change log ItemName simple type](#) 36
[change log ListId simple type](#) 18
[change log SiteId simple type](#) 41
[change log TimeLastModified simple type](#) 33
[change log WebId simple type](#) 42
[common](#) 18

Data types - simple

[change log ContentTypeId](#) 30
[change log DocId](#) 23
[change log Guid0](#) 24
[change log Int0](#) 27
[change log Int1](#) 38
[change log ItemFullUrl](#) 31
[change log ItemId](#) 21
[change log ItemName](#) 36
[change log ListId](#) 18
[change log SiteId](#) 41
[change log TimeLastModified](#) 33
[change log WebId](#) 42

Delete flag structure

[Deleted Documents result set](#) 71
[Distribution List E-mail Address result set](#) 46
[Document Content Stream Transfer result set](#) 47
[Document flag structure](#) 44
[Document Stream Storage Migration result set](#) 48
[Document Version Content Stream Transfer result set](#) 47
[Document Version Stream Storage Migration result set](#) 49

E

[Elements - overview](#) 52

Enumerations

[relationship delete behavior type](#) 46
[Event object type flag structure](#) 42
[Event type flag structure](#) 43

Events

[local - client](#) 229
[local - server](#) 228
[timer - client](#) 229
[timer - server](#) 228

Examples

[add just enough fields to cause allocation of additional rows per list item](#) 234
[allocate new rows while inserting an item into a wide list](#) 235
[cascade delete relationship behavior](#) 236
[change log](#) 230

[file fragment update behavior](#) 237
[link fixup](#) 231
[overview](#) 230
[themes](#) 233

F

[Fields - vendor-extensible](#) 17
[File fragment operations](#) 17
 [server](#) 57
[File Fragment result set](#) 49
[File fragment update behavior example](#) 237
Flag structures
 [delete](#) 44
 [document](#) 44
 [event object type](#) 42
 [event type](#) 43
 [security change type](#) 44
[fn_RoundDateToNearestSecond method](#) 58
[fn_UnpackCsvString method](#) 58

G

[Glossary](#) 10
[Groups - overview](#) 52

H

[Historical versioning operations](#) 16
 [server](#) 54

I

[Implementer - security considerations](#) 238
[Index of security parameters](#) 238
[Informative references](#) 15
Initialization
 [client](#) 228
 [server](#) 57
Interfaces - client
 [WSSDLIM](#) 228
[Introduction](#) 10

L

[Link fixup example](#) 231
[Link fixup operations](#) 16
 [server](#) 54
[List Data Source - complex type](#) 52
Local events
 [client](#) 229
 [server](#) 228
[Lookup relationships operations](#) 16
 [server](#) 55

M

Message processing
 [client](#) 229
 [server](#) 57
Messages
 [AllUserData table structure](#) 50

[attribute groups](#) 52
[attributes](#) 52
[binary structures](#) 46
[collate NameValuePair tables](#) 51
[common data types](#) 18
[delete flag structure](#) 44
[Distribution List E-mail Address result set](#) 46
[Document Content Stream Transfer result set](#) 47
[document flag structure](#) 44
[Document Stream Storage Migration result set](#) 48
[Document Version Content Stream Transfer result set](#) 47
[Document Version Stream Storage Migration result set](#) 49
[elements](#) 52
[event object type flag structure](#) 42
[event type flag structure](#) 43
[File Fragment result set](#) 49
[groups](#) 52
[List Data Source complex type](#) 52
[namespaces](#) 51
[NameValuePair table structure](#) 50
[NameValuePair Latin1 General CI AS table structure](#) 51
[security change type flag structure](#) 44
[simple types](#) 51
[Site Collection Flags result set](#) 46
[transport](#) 18

Methods

[fn_RoundDateToNearestSecond](#) 58
[fn_UnpackCsvString](#) 58
[proc_AddDependency](#) 58
[proc_AddEventToCache](#) 59
[proc_AddGhostDocument](#) 60
[proc_AddNewRowOrdToList](#) 63
[proc_AddNewRowOrdToListItem](#) 64
[proc_AL](#) 65
[proc_AppendFileFragmentDataById](#) 67
[proc_CascadeDeleteItems](#) 68
[proc_CheckIfExistingFieldHasDuplicateValues](#) 71
[proc_CheckIfExistingLookupsHaveValidParents](#) 72
[proc_CheckIfRestoreNeedsTranLock](#) 73
[proc_CheckoutDocumentInternal](#) 74
[proc_CloneDoc](#) 77
[proc_ConvertJunctionToLookup](#) 78
[proc_ConvertLookupToJunction](#) 79
[proc_CopyUrl](#) 79
[proc_CreateList](#) 84
[proc_CreateSharedAccessRequest](#) 88
[proc_CreateSite](#) 88
[proc_CreateView](#) 92
[proc_CreateWeb](#) 93
[proc_DeleteAllItemVersions](#) 97
[proc_DeleteAttachment](#) 98
[proc_DeleteAttachmentsFolder](#) 98
[proc_DeleteChanges](#) 99
[proc_DeleteEventLog](#) 100
[proc_DeleteFileFragmentsById](#) 215
[proc_DeleteFileFragmentsByTag](#) 100
[proc_DeleteFileFragmentsForCleanup](#) 216
[proc_DeleteFileFragmentsForUserDataDelete](#) 217

[proc_DeleteItemVersion](#) 101
[proc_DeleteSite](#) 102
[proc_DeleteSiteAsync](#) 103
[proc_DeleteSiteCoreAsync](#) 104
[proc_DeleteView](#) 105
[proc_DeleteWeb](#) 106
[proc_DirtyDocWithForwardLinks](#) 110
[proc_DropListRecord](#) 110
[proc_DropListUniqueField](#) 112
[proc_DropLookupRelationship](#) 113
[proc_EnsureSiteRbsCollection](#) 225
[proc_EnsureTranLockNotRequired](#) 114
[proc_FetchFileFormatMetaInfo](#) 114
[proc_FetchOldDoc](#) 115
[proc_FileFragmentPermissionCheck](#) 126
[proc_FindDocs](#) 127
[proc_FinishUndirtyList](#) 129
[proc_GenerateUniqueFileName](#) 129
[proc_GetAllAttachmentsInfo](#) 130
[proc_GetAllListsPlusProperties](#) 131
[proc_GetChanges](#) 132
[proc_GetChangeToken](#) 135
[proc_GetCurrent](#) 135
[proc_GetDocIdUrl](#) 136
[proc_GetDocStreamsForExternalMigration](#) 218
[proc_GetDocStreamsForInlineMigration](#) 218
[proc_GetDocStreamsForRbsMigration](#) 219
[proc_GetDocVersionStreamsForExternalMigration](#) 221
[proc_GetDocVersionStreamsForInlineMigration](#) 222
[proc_GetDocVersionStreamsForRbsMigration](#) 222
[proc_GetFileFragmentsById](#) 137
[proc_GetFileFragmentsByTag](#) 138
[proc_GetFullLinkInfoForSingleDoc](#) 139
[proc_GetListDataLinks](#) 142
[proc_GetListDataSource](#) 146
[proc_GetListItemsTreeToDelete](#) 147
[proc_GetNewListItemId](#) 149
[proc_GetRelatedFieldsForList](#) 150
[proc_GetSiteDeletionBatch](#) 151
[proc_GetStreamsForBackup](#) 224
[proc_GetStreamsForPreSiteMove](#) 226
[proc_GetStreamsForSiteMove](#) 224
[proc_GetUrlDocId](#) 152
[proc_GetWebUrlFromId](#) 152
[proc_InsertEventSubscriptionJunctionEntries](#) 153
[proc_InsertItemIntoNameValuePair](#) 166
[proc_InsertItemIntoNameValuePairCollated](#) 168
[proc_InsertJunction](#) 169
[proc_InsertListUniqueField](#) 170
[proc_InsertLookupRelationship](#) 170
[proc_IsFieldALookupRelationship](#) 171
[proc_IsSiteScheduledForDeletion](#) 172
[proc_ListRbsStoresWithIds](#) 226
[proc_ListThemeFiles](#) 172
[proc_ListThemes](#) 174
[proc_LoadTheme](#) 175
[proc_LogChange](#) 178
[proc_LogChangeForFileFragments](#) 179
[proc_PatchLinkForFile](#) 180

[proc_PatchLinkForWeb](#) 181
[proc_ReadSharedAccessRequests](#) 182
[proc_RecycleWeb](#) 108
[proc_RefreshCheckout](#) 183
[proc_RemoveJunctions](#) 184
[proc_RemoveSharedAccessRequest](#) 184
[proc_RenameHostHeaderSite](#) 185
[proc_RenameSite](#) 185
[proc_RestoreSite](#) 104
[proc_SaveFileFormatMetaInfo](#) 186
[proc_SaveFileFragmentById](#) 187
[proc_SaveFileFragmentByTag](#) 188
[proc_SetListDataSource](#) 189
[proc_SetNextId](#) 190
[proc_ShallowCopyRbsBlobs](#) 227
[proc_StartUndirtyList](#) 190
[proc_TakeOfflineDocument](#) 191
[proc_TranExtendLockWeb](#) 194
[proc_TranLockWeb](#) 195
[proc_TranUnlockWeb](#) 195
[proc_UndirtyListItem](#) 193
[proc_UpdateDirtyDocument](#) 196
[proc_UpdateEntityIdForList](#) 194
[proc_UpdateItemInNameValuePair](#) 197
[proc_UpdateItemInNameValuePairCollated](#) 199
[proc_UpdateLookupRelationship](#) 200
[proc_UpdateOrderNumber](#) 201
[proc_UpdateVersionVirusInfo](#) 202
[proc_UpdateView](#) 203
[proc_UpdateVirusInfo](#) 205
[proc_UpdateWebPartLinks](#) 206
[proc_UserHasDataItems](#) 207
[proc_ValidateLookupParents](#) 208
[proc_ValidateUniqueFields](#) 210
[proc_ValidateUniqueFieldsCollated](#) 212
[proc_VerifyListItemsTreeToDelete](#) 214

N

[Namespaces](#) 51
[NameValuePair](#) table structure 50
[NameValuePair Latin1 General CI AS table structure](#) 51
[Normative references](#) 15

O

[Overview \(synopsis\)](#) 16

P

[Parameters - security index](#) 238
[Preconditions](#) 17
[Prerequisites](#) 17
[proc_AddDependency method](#) 58
[proc_AddEventToCache method](#) 59
[proc_AddGhostDocument method](#) 60
[proc_AddNewRowOrdToList method](#) 63
[proc_AddNewRowOrdToListItem method](#) 64
[proc_AL method](#) 65
[proc_AppendFileFragmentDataById method](#) 67
[proc_CascadeDeleteItems](#)

[deleted documents result set](#) 71
[proc_CascadeDeleteItems method](#) 68
[proc_CheckIfExistsFieldHasDuplicateValues method](#) 71
[proc_CheckIfExistsLookupsHaveValidParents method](#) 72
[proc_CheckIfRestoreNeedsTranLock method](#) 73
[proc_CheckoutDocumentInternal method](#) 74
[proc_CloneDoc method](#) 77
[proc_ConvertJunctionToLookup method](#) 78
[proc_ConvertLookupToJunction method](#) 79
[proc_CopyUrl method](#) 79
[proc_CreateList method](#) 84
[proc_CreateSharedAccessRequest method](#) 88
[proc_CreateSite method](#) 88
[proc_CreateView method](#) 92
[proc_CreateWeb method](#) 93
[proc_DeleteAllItemVersions method](#) 97
[proc_DeleteAttachment method](#) 98
[proc_DeleteAttachmentsFolder method](#) 98
[proc_DeleteChanges method](#) 99
[proc_DeleteEventLog method](#) 100
[proc_DeleteFileFragmentsById method](#) 215
[proc_DeleteFileFragmentsByTag method](#) 100
[proc_DeleteFileFragmentsForCleanup method](#) 216
[proc_DeleteFileFragmentsForUserDataDelete method](#) 217
[proc_DeleteItemVersion method](#) 101
[proc_DeleteSite method](#) 102
[proc_DeleteSiteAsync method](#) 103
[proc_DeleteSiteCoreAsync method](#) 104
[proc_DeleteView method](#) 105
[proc_DeleteWeb method](#) 106
[proc_DirtyDocWithForwardLinks method](#) 110
[proc_DropListRecord method](#) 110
[proc_DropListUniqueField method](#) 112
[proc_DropLookupRelationship method](#) 113
[proc_EnsureSiteRbsCollection method](#) 225
[proc_EnsureTranLockNotRequired method](#) 114
[proc_FetchFileFormatMetaInfo method](#) 114
[proc_FetchOldDoc method](#) 115
[proc_FileFragmentPermissionCheck method](#) 126
[proc_FindDocs method](#) 127
[proc_FinishUndirtyList method](#) 129
[proc_GenerateUniqueFileName method](#) 129
[proc_GetAllAttachmentsInfo method](#) 130
[proc_GetAllListsPlusProperties method](#) 131
[proc_GetChanges method](#) 132
[proc_GetChangeToken method](#) 135
[proc_GetCurrent method](#) 135
[proc_GetDocIdUrl method](#) 136
[proc_GetDocStreamsForExternalMigration method](#) 218
[proc_GetDocStreamsForInlineMigration method](#) 218
[proc_GetDocStreamsForRbsMigration method](#) 219
[proc_GetDocVersionStreamsForExternalMigration method](#) 221
[proc_GetDocVersionStreamsForInlineMigration method](#) 222
[proc_GetDocVersionStreamsForRbsMigration method](#) 222
[proc_GetFileFragmentsById method](#) 137
[proc_GetFileFragmentsByTag method](#) 138
[proc_GetFullLinkInfoForSingleDoc method](#) 139
[proc_GetListDataLinks method](#) 142
[proc_GetListDataSource method](#) 146
[proc_GetListItemsTreeToDelete method](#) 147
[proc_GetNewListItemId method](#) 149
[proc_GetRelatedFieldsForList method](#) 150
[proc_GetSiteDeletionBatch method](#) 151
[proc_GetStreamsForBackup method](#) 224
[proc_GetStreamsForPreSiteMove method](#) 226
[proc_GetStreamsForSiteMove method](#) 224
[proc_GetUrlDocId method](#) 152
[proc_GetWebUrlFromId method](#) 152
[proc_InsertEventSubscriptionJunctionEntries method](#) 153
[proc_InsertItemIntoNameValuePair method](#) 166
[proc_InsertItemIntoNameValuePairCollated method](#) 168
[proc_InsertJunction method](#) 169
[proc_InsertListUniqueField method](#) 170
[proc_InsertLookupRelationship method](#) 170
[proc_IsFieldALookupRelationship method](#) 171
[proc_IsSiteScheduledForDeletion method](#) 172
[proc_ListRbsStoresWithIds method](#) 226
[proc_ListThemeFiles method](#) 172
[proc_ListThemes method](#) 174
[proc_LoadTheme method](#) 175
[proc_LogChange method](#) 178
[proc_LogChangeForFileFragments method](#) 179
[proc_PatchLinkForFile method](#) 180
[proc_PatchLinkForWeb method](#) 181
[proc_ReadSharedAccessRequests method](#) 182
[proc_RecycleWeb method](#) 108
[proc_RefreshCheckout method](#) 183
[proc_RemoveJunctions method](#) 184
[proc_RemoveSharedAccessRequest method](#) 184
[proc_RenameHostHeaderSite method](#) 185
[proc_RenameSite method](#) 185
[proc_RestoreSite method](#) 104
[proc_SaveFileFormatMetaInfo method](#) 186
[proc_SaveFileFragmentById method](#) 187
[proc_SaveFileFragmentByTag method](#) 188
[proc_SetListDataSource method](#) 189
[proc_SetNextId method](#) 190
[proc_ShallowCopyRbsBlobs method](#) 227
[proc_StartUndirtyList method](#) 190
[proc_TakeOfflineDocument method](#) 191
[proc_TranExtendLockWeb method](#) 194
[proc_TranLockWeb method](#) 195
[proc_TranUnlockWeb method](#) 195
[proc_UndirtyListItem method](#) 193
[proc_UpdateDirtyDocument method](#) 196
[proc_UpdateEntityIdForList method](#) 194
[proc_UpdateItemInNameValuePair method](#) 197
[proc_UpdateItemInNameValuePairCollated method](#) 199
[proc_UpdateLookupRelationship method](#) 200
[proc_UpdateOrderNumber method](#) 201
[proc_UpdateVersionVirusInfo method](#) 202
[proc_UpdateView method](#) 203

[proc_UpdateVirusInfo method](#) 205
[proc_UpdateWebPartLinks method](#) 206
[proc_UserHasDataItems method](#) 207
[proc_ValidateLookupParents method](#) 208
[proc_ValidateUniqueFields method](#) 210
[proc_ValidateUniqueFieldsCollated method](#) 212
[proc_VerifyListItemsTreeToDelete method](#) 214
[Product behavior](#) 239
[Publish and un-publish operations](#) 16
[server](#) 53

R

[References](#) 14
[informative](#) 15
[normative](#) 15
[Relationship delete behavior type enumeration](#) 46
[Relationship to other protocols](#) 17
[Result set](#)
[proc_CascadeDeleteItems - deleted documents](#) 71
[Result sets - messages](#)
[Distribution List E-mail Address](#) 46
[Document Content Stream Transfer](#) 47
[Document Stream Storage Migration](#) 48
[Document Version Content Stream Transfer](#) 47
[Document Version Stream Storage Migration](#) 49
[File Fragment](#) 49
[Site Collection Flags](#) 46

S

[Security](#)
[implementer considerations](#) 238
[parameter index](#) 238
[Security change type flag structure](#) 44
[Sequencing rules](#)
[client](#) 229
[server](#) 57
[Server](#)
[abstract data model](#) 53
[change log operations](#) 53
[check-in and check-out operations](#) 53
[file fragment operations](#) 57
[fn_RoundDateToNearestSecond method](#) 58
[fn_UnpackCsvString method](#) 58
[historical versioning operations](#) 54
[initialization](#) 57
[link fixup operations](#) 54
[local events](#) 228
[lookup relationships operations](#) 55
[message processing](#) 57
[proc_AddDependency method](#) 58
[proc_AddEventToCache method](#) 59
[proc_AddGhostDocument method](#) 60
[proc_AddNewRowOrdToList method](#) 63
[proc_AddNewRowOrdToListItem method](#) 64
[proc_AL method](#) 65
[proc_AppendFileFragmentDataById method](#) 67
[proc_CascadeDeleteItems method](#) 68
[proc_CheckIfExistingFieldHasDuplicateValues method](#) 71

[proc_CheckIfExistingLookupsHaveValidParents method](#) 72
[proc_CheckIfRestoreNeedsTranLock method](#) 73
[proc_CheckoutDocumentInternal method](#) 74
[proc_CloneDoc method](#) 77
[proc_ConvertJunctionToLookup method](#) 78
[proc_ConvertLookupToJunction method](#) 79
[proc_CopyUrl method](#) 79
[proc_CreateList method](#) 84
[proc_CreateSharedAccessRequest method](#) 88
[proc_CreateSite method](#) 88
[proc_CreateView method](#) 92
[proc_CreateWeb method](#) 93
[proc_DeleteAllItemVersions method](#) 97
[proc_DeleteAttachment method](#) 98
[proc_DeleteAttachmentsFolder method](#) 98
[proc_DeleteChanges method](#) 99
[proc_DeleteEventLog method](#) 100
[proc_DeleteFileFragmentsById method](#) 215
[proc_DeleteFileFragmentsByTag method](#) 100
[proc_DeleteFileFragmentsForCleanup method](#) 216
[proc_DeleteFileFragmentsForUserDataDelete method](#) 217
[proc_DeleteItemVersion method](#) 101
[proc_DeleteSite method](#) 102
[proc_DeleteSiteAsync method](#) 103
[proc_DeleteSiteCoreAsync method](#) 104
[proc_DeleteView method](#) 105
[proc_DeleteWeb method](#) 106
[proc_DirtyDocWithForwardLinks method](#) 110
[proc_DropListRecord method](#) 110
[proc_DropListUniqueField method](#) 112
[proc_DropLookupRelationship method](#) 113
[proc_EnsureSiteRbsCollection method](#) 225
[proc_EnsureTranLockNotRequired method](#) 114
[proc_FetchFileFormatMetaInfo method](#) 114
[proc_FetchOldDoc method](#) 115
[proc_FileFragmentPermissionCheck method](#) 126
[proc_FindDocs method](#) 127
[proc_FinishUndirtyList method](#) 129
[proc_GenerateUniqueFileName method](#) 129
[proc_GetAllAttachmentsInfo method](#) 130
[proc_GetAllListsPlusProperties method](#) 131
[proc_GetChanges method](#) 132
[proc_GetChangeToken method](#) 135
[proc_GetCurrent method](#) 135
[proc_GetDocIdUrl method](#) 136
[proc_GetDocStreamsForExternalMigration method](#) 218
[proc_GetDocStreamsForInlineMigration method](#) 218
[proc_GetDocStreamsForRbsMigration method](#) 219
[proc_GetDocVersionStreamsForExternalMigration method](#) 221
[proc_GetDocVersionStreamsForInlineMigration method](#) 222
[proc_GetDocVersionStreamsForRbsMigration method](#) 222
[proc_GetFileFragmentsById method](#) 137
[proc_GetFileFragmentsByTag method](#) 138
[proc_GetFullLinkInfoForSingleDoc method](#) 139

[proc_GetListDataLinks method](#) 142
[proc_GetListDataSource method](#) 146
[proc_GetListItemsTreeToDelete method](#) 147
[proc_GetNewListItemId method](#) 149
[proc_GetRelatedFieldsForList method](#) 150
[proc_GetSiteDeletionBatch method](#) 151
[proc_GetStreamsForBackup method](#) 224
[proc_GetStreamsForPreSiteMove method](#) 226
[proc_GetStreamsForSiteMove method](#) 224
[proc_GetUrlDocId method](#) 152
[proc_GetWebUrlFromId method](#) 152
[proc_InsertEventSubscriptionJunctionEntries method](#) 153
[proc_InsertItemIntoNameValuePair method](#) 166
[proc_InsertItemIntoNameValuePairCollated method](#) 168
[proc_InsertJunction method](#) 169
[proc_InsertListUniqueField method](#) 170
[proc_InsertLookupRelationship method](#) 170
[proc_IsFieldALookupRelationship method](#) 171
[proc_IsSiteScheduledForDeletion method](#) 172
[proc_ListRbsStoresWithIds method](#) 226
[proc_ListThemeFiles method](#) 172
[proc_ListThemes method](#) 174
[proc_LoadTheme method](#) 175
[proc_LogChange method](#) 178
[proc_LogChangeForFileFragments method](#) 179
[proc_PatchLinkForFile method](#) 180
[proc_PatchLinkForWeb method](#) 181
[proc_ReadSharedAccessRequests method](#) 182
[proc_RecycleWeb method](#) 108
[proc_RefreshCheckout method](#) 183
[proc_RemoveJunctions method](#) 184
[proc_RemoveSharedAccessRequest method](#) 184
[proc_RenameHostHeaderSite method](#) 185
[proc_RenameSite method](#) 185
[proc_RestoreSite method](#) 104
[proc_SaveFileFormatMetaInfo method](#) 186
[proc_SaveFileFragmentById method](#) 187
[proc_SaveFileFragmentByTag method](#) 188
[proc_SetListDataSource method](#) 189
[proc_SetNextId method](#) 190
[proc_ShallowCopyRbsBlobs method](#) 227
[proc_StartUndirtyList method](#) 190
[proc_TakeOfflineDocument method](#) 191
[proc_TranExtendLockWeb method](#) 194
[proc_TranLockWeb method](#) 195
[proc_TranUnlockWeb method](#) 195
[proc_UndirtyListItem method](#) 193
[proc_UpdateDirtyDocument method](#) 196
[proc_UpdateEntityIdForList method](#) 194
[proc_UpdateItemInNameValuePair method](#) 197
[proc_UpdateItemInNameValuePairCollated method](#) 199
[proc_UpdateLookupRelationship method](#) 200
[proc_UpdateOrderNumber method](#) 201
[proc_UpdateVersionVirusInfo method](#) 202
[proc_UpdateView method](#) 203
[proc_UpdateVirusInfo method](#) 205
[proc_UpdateWebPartLinks method](#) 206
[proc_UserHasDataItems method](#) 207

[proc_ValidateLookupParents method](#) 208
[proc_ValidateUniqueFields method](#) 210
[proc_ValidateUniqueFieldsCollated method](#) 212
[proc_VerifyListItemsTreeToDelete method](#) 214
[publish and un-publish operations](#) 53
[sequencing rules](#) 57
[theme operations](#) 55
[timer events](#) 228
[timers](#) 57
[wide list operations](#) 57

Simple types

[change_log_ContentTypeId](#) 30
[change_log_DocId](#) 23
[change_log_Guid0](#) 24
[change_log_Int0](#) 27
[change_log_Int1](#) 38
[change_log_ItemFullUrl](#) 31
[change_log_ItemId](#) 21
[change_log_ItemName](#) 36
[change_log_ListId](#) 18
[change_log_SiteId](#) 41
[change_log_TimeLastModified](#) 33
[change_log_WebId](#) 42

Simple types - overview

[Site Collection Flags result set](#) 46

[Standards assignments](#) 17

Structures

[binary](#) 46

T

Table structures

[AllUserData](#) 50

[collated NameValuePair](#) 51

[NameValuePair](#) 50

[NameValuePair_Latin1_General_CI_AS](#) 51

[Theme operations](#) 16

[server](#) 55

[Themes example](#) 233

Timer events

[client](#) 229

[server](#) 228

Timers

[client](#) 228

[server](#) 57

[Tracking changes](#) 240

[Transport](#) 18

Types

[simple](#) 51

V

[Vendor-extensible fields](#) 17

[Versioning](#) 17

W

[Wide list operations](#) 16

[server](#) 57

[WSSDLIM interface](#) 228