[MS-WSSFO]:
Windows SharePoint Services (WSS):
File Operations Database Communications Protocol

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. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names.** The example companies, organizations, products, domain names, email 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.

Copyright © 2014 Microsoft Corporation.

Revision Summary

Date	Revision History	Revision Class	Comments
03/14/2008	0.1	Major	Initial Availability.
06/20/2008	0.1.1	Editorial	Revised and edited the technical content.
07/25/2008	0.1.2	Editorial	Revised and edited the technical content.
08/29/2008	1.0	Major	Updated and revised the technical content.
10/24/2008	1.0.1	Editorial	Revised and edited the technical content.
12/05/2008	1.0.2	Editorial	Initial availability
01/16/2009	1.0.3	Editorial	Revised and edited the technical content.
02/27/2009	1.0.4	Editorial	Revised and edited the technical content.
04/10/2009	2.0	Major	Updated and revised the technical content.
05/22/2009	2.1	Minor	Updated the technical content.
07/02/2009	3.0	Major	Updated and revised the technical content.
08/14/2009	3.1	Minor	Updated the technical content.
09/25/2009	3.2	Minor	Updated the technical content.
11/06/2009	3.2.1	Editorial	Revised and edited the technical content.
12/18/2009	4.0	Major	Updated and revised the technical content.
01/29/2010	4.0.1	Editorial	Revised and edited the technical content.
03/12/2010	4.0.2	Editorial	Revised and edited the technical content.
04/23/2010	4.0.3	Editorial	Revised and edited the technical content.
06/04/2010	4.1	Minor	Updated the technical content.
07/16/2010	4.1	No change	No changes to the meaning, language, or formatting of the technical content.
08/27/2010	4.1	No change	No changes to the meaning, language, or formatting of the technical content.
10/08/2010	4.1	No change	No changes to the meaning, language, or formatting of the technical content.
11/19/2010	5.0	Major	Significantly changed the technical content.
01/07/2011	5.0	No change	No changes to the meaning, language, or formatting of the technical content.
02/11/2011	5.0	No change	No changes to the meaning, language, or formatting of

Copyright © 2014 Microsoft Corporation.

Date	Revision History	Revision Class	Comments
			the technical content.
03/25/2011	5.0	No change	No changes to the meaning, language, or formatting of the technical content.
05/06/2011	5.0	No change	No changes to the meaning, language, or formatting of the technical content.
06/17/2011	5.1	Minor	Clarified the meaning of the technical content.
09/23/2011	6.0	Major	Significantly changed the technical content.
12/16/2011	6.0	No change	No changes to the meaning, language, or formatting of the technical content.
03/30/2012	6.0	No change	No changes to the meaning, language, or formatting of the technical content.
07/12/2012	6.0	No change	No changes to the meaning, language, or formatting of the technical content.
09/12/2012	6.0	No change	No changes to the meaning, language, or formatting of the technical content.
10/08/2012	6.0	No change	No changes to the meaning, language, or formatting of the technical content.
02/11/2013	6.0	No change	No changes to the meaning, language, or formatting of the technical content.
07/30/2013	6.0	No change	No changes to the meaning, language, or formatting of the technical content.
11/18/2013	6.0	No change	No changes to the meaning, language, or formatting of the technical content.
02/10/2014	6.0	No change	No changes to the meaning, language, or formatting of the technical content.
04/30/2014	6.1	Minor	Clarified the meaning of the technical content.
07/31/2014	7.0	Major	Significantly changed the technical content.
10/30/2014	7.1	Minor	Clarified the meaning of the technical content.

Copyright © 2014 Microsoft Corporation.

Table of Contents

1 Introduction 16
1.1 Glossary 16
1.2 References 18
1.2.1 Normative References 18
1.2.2 Informative References 19
1.3 Overview 19
1.3.1 File Operations 19
1.3.2 User and Group Operations 19
1.4 Relationship to Other Protocols 19
1.5 Prerequisites/Preconditions 20
1.6 Applicability Statement 20
1.7 Versioning and Capability Negotiation 20
1.8 Vendor-Extensible Fields 20
1.9 Standards Assignments 20

2 Messages 21
2.1 Transport 21
2.2 Message Syntax 21
2.2.1 Simple Data Types 21
2.2.1.1 Content Type Identifier 21
2.2.1.2 Document Identifier 21
2.2.1.3 Event Receiver Identifier 21
2.2.1.4 Feature Identifier 21
2.2.1.5 List Identifier 21
2.2.1.6 List Item Identifier 21
2.2.1.7 Role Identifier 22
2.2.1.8 Scope Identifier 22
2.2.1.9 Site Collection Identifier 22
2.2.1.10 Site Group Identifier 22
2.2.1.11 Site Identifier 22
2.2.1.12 SystemID 22
2.2.1.13 User Identifier 22
2.2.1.14 View Identifier 22
2.2.1.15 Web Part Identifier 22
2.2.1.16 Workflow Identifier 22
2.2.2 Bit Fields and Flag Structures 23
2.2.2.1 Audit Flags 23
2.2.2.2 Configuration Object Status 23
2.2.2.3 Doc Flags 24
2.2.2.4 Document Store Type 25
2.2.2.5 List Flags 25
2.2.2.6 Publishing Level Type 27
2.2.2.7 Put Flags Type 28
2.2.2.8 Rename Flags 29
2.2.2.9 Site Collection Flags 30
2.2.2.10 Site Property Flags 30
2.2.2.11 View Flags 31
2.2.2.12 Workdays Flag 32
2.2.2.13 WSS Rights Mask 33
2.2.3 Enumerations 35

Copyright © 2014 Microsoft Corporation.

2.2.3.1 Attachments Flag 35
2.2.3.2 Audit Item Type 36
2.2.3.3 Calendar Type 36
2.2.3.4 Collation Order Enumeration 37
2.2.3.5 Event Host Type 38
2.2.3.6 Event Receiver Type 39
2.2.3.7 Excluded Folder Type 40
2.2.3.8 LinkDynamic Type 40
2.2.3.9 LinkSecurity Type 41
2.2.3.10 LinkType Types 41
2.2.3.11 List Base Type 42
2.2.3.12 List Server Template 42
2.2.3.13 Moderation Status 44
2.2.3.14 Page Type 44
2.2.3.15 Redirect Type 45
2.2.3.16 Role Definition Type 45
2.2.3.17 Time Zone Identifier 46
2.2.3.18 Virus Status 49
2.2.4 Binary Structures 49
2.2.4.1 Calendar View Options Type 49
2.2.4.2 External Group Token 50
2.2.4.3 Token Group Offset and Attributes 51
2.2.4.4 Token Groups 52
2.2.4.5 WSS ACE 52
2.2.4.6 WSS ACL Format 53
2.2.4.7 WSS External Group Map Cache Format 53
2.2.4.8 WSS External Group Record 54
2.2.4.9 WSS User Token 54
2.2.5 Result Sets 55
2.2.5.1 ACL and Permission Result Set 55
2.2.5.2 Attachment State Result Set 55
2.2.5.3 Domain Group Cache BEDS Update Result Set 56
2.2.5.4 Domain Group Cache Versions Result Set 56
2.2.5.5 Domain Group Cache WFE Update Result Set 57
2.2.5.6 Document Metadata Result Set 57
2.2.5.7 Document Version Metadata Result Set 60
2.2.5.8 Empty Result Set 64
2.2.5.9 Event Receivers Result Set 64
2.2.5.10 Individual URL Security Result Set 66
2.2.5.11 Link Information Result Set 67
2.2.5.12 List Metadata Result Set 68
2.2.5.13 List Web Parts Result Set 72
2.2.5.14 NULL Individual URL Security Result Set 73
2.2.5.15 NULL Unique Permissions Result Set 74
2.2.5.16 Object ID Result Set 74
2.2.5.17 Principal User Information Result Set 74
2.2.5.18 Server Time Result Set 75
2.2.5.19 Single Doc Link Information Result Set 75
2.2.5.20 Site Audit Mask Result Set 76
2.2.5.21 Site Feature List Result Set 77
2.2.5.22 Site Categories Result Set 77
2.2.5.23 Site Metadata Result Set 77
2.2.5.24 Site MetaInfo Result Set 82

Copyright © 2014 Microsoft Corporation.

2.2.5.25 Unique Permissions Result Set 83
2.2.5.26 URL Result Set 83
2.2.6 SQL Structures 83
2.2.6.1 Configuration Object 83
2.2.6.1.1 Class 83
2.2.6.1.2 Id 84
2.2.6.1.3 Name 84
2.2.6.1.4 Parent 84
2.2.6.1.5 Status 85
2.2.6.1.6 Version 85
2.2.6.1.7 Properties 85
2.2.6.1.7.1 Alternate URL Collection 85
2.2.6.1.7.2 Content Database 86
2.2.6.1.7.3 Web Application 86
2.2.6.2 Dependencies 87
2.2.7 Tables and Views 87
2.2.7.1 AllDocs Table 87
2.2.7.2 AllDocStreams Table 93
2.2.7.3 AllUserData Table 94
2.2.7.4 Docs View 99
2.2.7.5 GroupMembership Table 100
2.2.7.6 Sec_SiteGroupsView 100
2.2.7.7 Sites Table 103
2.2.7.8 UserData View 106
2.2.7.9 UserDataVersioned View 106
2.2.7.10 UserInfo Table 106
2.2.7.11 Versions 108
2.2.8 XML Structures 109
2.2.8.1 Namespaces 109
2.2.8.2 Simple Types 109
2.2.8.2.1 FALSE_Case_Insensitive_Else_Anything 109
2.2.8.2.2 FieldAggregationAttribute 110
2.2.8.2.3 FieldInternalType 111
2.2.8.2.4 FieldRefType 112
2.2.8.2.5 FieldRichTextMode 112
2.2.8.2.6 IMEMode 113
2.2.8.2.7 IntPositive 113
2.2.8.2.8 Join Type 114
2.2.8.2.9 TextDirection 114
2.2.8.2.10 TRUE_If_Present 115
2.2.8.2.11 TRUEFALSE 115
2.2.8.2.12 UniqueIdentifierWithoutBraces 115
2.2.8.2.13 UniqueIdentifierWithoutBracesOrEmpty 115
2.2.8.2.14 UniqueIdentifierWithOrWithoutBraces 115
2.2.8.3 Complex Types 116
2.2.8.3.1 CHOICEDEFINITION Type 116
2.2.8.3.1.1 Schema 116
2.2.8.3.1.2 Attributes 116
2.2.8.3.1.3 Child Elements 116
2.2.8.3.2 CHOICEDEFINITIONS Type 116
2.2.8.3.2.1 Schema 116
2.2.8.3.2.2 Attributes 116
2.2.8.3.2.3 Child Elements 116

Copyright © 2014 Microsoft Corporation.

2.2.8.3.3 FieldDefinition Type 117
2.2.8.3.3.1 Schema 117
2.2.8.3.3.2 Attributes 120
2.2.8.3.3.3 Child Elements 128
2.2.8.3.4 FieldDefinitionDatabase Type 129
2.2.8.3.4.1 Schema 129
2.2.8.3.4.2 Attributes 129
2.2.8.3.4.3 Child Elements 129
2.2.8.3.5 FieldDefinitionDatabaseWithVersion Type 129
2.2.8.3.5.1 Schema 129
2.2.8.3.5.2 Attributes 129
2.2.8.3.5.3 Child Elements 129
2.2.8.3.6 FieldDefinitionTP Type 130
2.2.8.3.6.1 Schema 130
2.2.8.3.6.2 Attributes 130
2.2.8.3.6.3 Child Elements 130
2.2.8.3.7 FieldParserRef Type 130
2.2.8.3.7.1 Schema 130
2.2.8.3.7.2 Attributes 130
2.2.8.3.7.3 Child Elements 131
2.2.8.3.8 FieldParserRefs Type 131
2.2.8.3.8.1 Schema 131
2.2.8.3.8.2 Attributes 131
2.2.8.3.8.3 Child Elements 131
2.2.8.3.9 FieldRefDefinitionField Type 131
2.2.8.3.9.1 Schema 131
2.2.8.3.9.2 Attributes 131
2.2.8.3.9.3 Child Elements 132
2.2.8.3.10 FieldRefDefinitionTP Type 132
2.2.8.3.10.1 Schema 132
2.2.8.3.10.2 Attributes 132
2.2.8.3.10.3 Child Elements 132
2.2.8.3.11 MAPPINGDEFINITION Type 132
2.2.8.3.11.1 Schema 132
2.2.8.3.11.2 Attributes 133
2.2.8.3.11.3 Child Elements 133
2.2.8.3.12 MAPPINGDEFINITIONS Type 133
2.2.8.3.12.1 Schema 133
2.2.8.3.12.2 Attributes 133
2.2.8.3.12.3 Child Elements 133
2.2.8.4 Elements 133
2.2.8.5 Attributes 133
2.2.8.6 Groups 133
2.2.8.7 Attribute Groups 134

3 Protocol Details 135
3.1 Back End Database Server Details 135
3.1.1 Abstract Data Model 135
3.1.2 Timers 136
3.1.3 Initialization 136
3.1.4 Higher-Layer Triggered Events 136
3.1.5 Message Processing Events and Sequencing Rules 136
3.1.5.1 fn_GetFullUrl 140

Copyright © 2014 Microsoft Corporation.

3.1.5.2 proc_AddBuildDependency 141
3.1.5.3 proc_AddDocument 141
3.1.5.3.1 Site List for Normalization Result Set 145
3.1.5.3.2 Checkout Information Result Set 145
3.1.5.4 proc_AddListItem 145
3.1.5.5 proc_ChangeLevelForDoc 154
3.1.5.6 proc_CheckoutDocument 156
3.1.5.6.1 Link Info Single Doc Result Set 157
3.1.5.6.2 Document Metadata Result Set 158
3.1.5.6.3 NULL Result Set 158
3.1.5.6.4 Event Receivers Result Set 158
3.1.5.6.5 Audit Mask Result Set 158
3.1.5.7 proc_ClearLinks 158
3.1.5.8 proc_CreateDir 159
3.1.5.9 proc_DeleteAllDocumentVersions 161
3.1.5.10 proc_DeleteDocBuildDependencySet 162
3.1.5.11 proc_DeleteDocumentVersion 163
3.1.5.12 proc_DeleteUrl 164
3.1.5.12.1 Deleted Documents Result Set 166
3.1.5.12.2 Deleted Aliased Lists Result Set 166
3.1.5.12.3 Empty Deleted Aliased Lists Result Set 166
3.1.5.13 proc_DirtyDependents 167
3.1.5.14 proc_EnumLists 167
3.1.5.14.1 List Information Result Set 169
3.1.5.14.2 Recycle Bin Information Result Set 172
3.1.5.14.3 NULL Result Set 173
3.1.5.15 proc_FetchDocForHttpGet 173
3.1.5.15.1 HTTP Document Metadata Result Set 176
3.1.5.15.2 Domain Group Cache Versions Result Set (1) 179
3.1.5.15.3 Domain Group Cache BEDS Update Result Set 179
3.1.5.15.4 Domain Group Cache WFE Update Result Set 179
3.1.5.15.5 User Information Result Set 179
3.1.5.15.6 Welcome Page Redirect Information Result Set 180
3.1.5.15.7 Non-Welcome Page Redirect Information Result Set 181
3.1.5.15.8 Document Content Stream Result Set 181
3.1.5.15.9 Site Collection Audit Mask Result Set 182
3.1.5.15.10 List Audit Mask Result Set 182
3.1.5.15.11 Document Build Dependency Set Result Set 183
3.1.5.15.12 Document Build Dependency Metadata Result Set 183
3.1.5.15.13 Domain Group Cache Versions Result Set (2) 185
3.1.5.15.14 Site Metadata Result Set 185
3.1.5.15.15 Event Receivers Result Set 185
3.1.5.15.16 WebParts Metadata, Personalized Result Set 185
3.1.5.15.17 Web Parts Metadata, Nonpersonalized Result Set 187
3.1.5.15.18 List Metadata Result Set 188
3.1.5.15.19 List Event Receivers Result Set 192
3.1.5.15.20 List Security Information Result Set 193
3.1.5.15.21 List Web Parts Result Set 193
3.1.5.15.22 Content Type Order Result Set 193
3.1.5.15.23 Current Folder Scope Result Set 194
3.1.5.15.24 Site Features List Result Set 194
3.1.5.15.25 Navigation Context Security Information Result Set 194
3.1.5.15.26 NULL Navigation Context Security Information Result Set 195

Copyright © 2014 Microsoft Corporation.

3.1.5.15.27 Empty Navigation Context Security Information Result Set 195
3.1.5.16 proc_FetchDocForRead 195
3.1.5.16.1 Subsite List Result Set 197
3.1.5.16.2 Link Info Single Doc Result Set 197
3.1.5.16.3 Document Metadata Result Set 197
3.1.5.16.4 Document Version Metadata Result Set 197
3.1.5.16.5 NULL Result Set 197
3.1.5.16.6 Event Receivers Result Set (1) 198
3.1.5.16.7 List Metadata Result Set 198
3.1.5.16.8 Empty List Result Set 198
3.1.5.16.9 Event Receivers Result Set (2) 198
3.1.5.16.10 Document Information and Content (Read) Result Set 198
3.1.5.16.11 Document Version Information and Content Result Set 200
3.1.5.16.12 Attachment State Result Set 201
3.1.5.16.13 Audit Mask Result Set 201
3.1.5.17 proc_FetchDocForUpdate 201
3.1.5.17.1 Subsite List Result Set 202
3.1.5.17.2 ACL and Permission Result Set 203
3.1.5.17.3 Document Metadata Result Set 203
3.1.5.17.4 Document Version Metadata Result Set 203
3.1.5.17.5 NULL Result Set 203
3.1.5.17.6 Event Receivers Result Set (1) 203
3.1.5.17.7 Link Info Single Doc Fixup Result Set 204
3.1.5.17.8 Web Part Info Result Set 205
3.1.5.17.9 Zone ID Result Set 205
3.1.5.17.10 List Metadata Result Set 205
3.1.5.17.11 Empty List Result Set 205
3.1.5.17.12 Event Receivers Result Set (2) 206
3.1.5.17.13 Document Information and Content (Update) Result Set 206
3.1.5.17.14 Document Version 1 Information and Content Result Set 207
3.1.5.17.15 Document Version 2 Information and Content Result Set 208
3.1.5.17.16 Attachment State Result Set 208
3.1.5.18 proc_FetchWelcomeNames 209
3.1.5.18.1 Welcome Pages Result Set 209
3.1.5.19 proc_GenerateNextId 209
3.1.5.20 proc_GetAllAttachmentsInfo 210
3.1.5.20.1 List Attachments Result Set 210
3.1.5.20.2 Document Attachments Result Set 211
3.1.5.21 proc_GetAuditMask 211
3.1.5.21.1 Audit Mask Result Set 211
3.1.5.22 proc_GetAuditMaskOutput 211
3.1.5.23 proc_GetBlobIdsDocs 212
3.1.5.23.1 Document Stream External Storage Information Result Set 213
3.1.5.24 proc_GetBlobIdsVersions 213
3.1.5.24.1 Document Version External Storage Information Result Set 213
3.1.5.25 proc_GetContainingList 214
3.1.5.25.1 List Metadata Result Set 214
3.1.5.25.2 Empty Result Set 215
3.1.5.25.3 Event Receivers Result Set 215
3.1.5.26 proc_GetDocsMetaInfo 215
3.1.5.26.1 Individual URL Security Result Set 216
3.1.5.26.2 NULL Individual Url Security Result Set 217
3.1.5.26.3 Server Time Result Set 217

Copyright © 2014 Microsoft Corporation.

3.1.5.26.4 Subsite List Result Set 217
3.1.5.26.5 Link Info Result Set 217
3.1.5.26.6 Multiple Document Metadata Result Set 218
3.1.5.27 proc_GetLinkInfoSingleDoc 221
3.1.5.27.1 Link Info Single Doc Result Set 222
3.1.5.28 proc_GetListCheckedOutFiles 222
3.1.5.28.1 Checked Out Files Result Set 222
3.1.5.29 proc_GetListFields 223
3.1.5.29.1 Fields Information Result Set 223
3.1.5.30 proc_GetListMetaDataAndEventReceivers 224
3.1.5.30.1 List Metadata Result Set 224
3.1.5.30.2 Unique Permissions Result Set 225
3.1.5.30.3 NULL Unique Permissions Result Set 225
3.1.5.30.4 List Event Receivers Result Set 225
3.1.5.30.5 List Web Parts Result Set 225
3.1.5.31 proc_getObject 225
3.1.5.31.1 Object Result Set 226
3.1.5.32 proc_getObjectsByBaseClass 226
3.1.5.32.1 Object ID Result Set 227
3.1.5.33 proc_getObjectsByClass 227
3.1.5.33.1 Object ID Result Set 227
3.1.5.34 proc_GetSiteFlags 227
3.1.5.34.1 Site Collection Flags Result Set 228
3.1.5.35 proc_getSiteMap 228
3.1.5.35.1 Site Map Result Set 228
3.1.5.36 proc_getSiteMapById 229
3.1.5.36.1 Site Map By Id Result Set 229
3.1.5.37 proc_GetTpWebMetaDataAndListMetaData 230
3.1.5.37.1 Web Url Result Set 233
3.1.5.37.2 Domain Group Cache Versions Result Set 233
3.1.5.37.3 Domain Group Cache BEDS Update Result Set 233
3.1.5.37.4 Domain Group Cache WFE Update Result Set 234
3.1.5.37.5 Site Metadata Result Set 234
3.1.5.37.6 Event Receivers Result Set (1) 234
3.1.5.37.7 Site Categories Result Set 234
3.1.5.37.8 Site MetaInfo Result Set 234
3.1.5.37.9 Site Feature List Result Set 235
3.1.5.37.10 Unique Permissions Result Set (1) 235
3.1.5.37.11 NULL Unique Permissions Result Set (1) 235
3.1.5.37.12 Empty Result Set 235
3.1.5.37.13 Redirect Url Result Set 236
3.1.5.37.14 No Welcome Redirect Url Result Set 236
3.1.5.37.15 List Identifier Result Set 237
3.1.5.37.16 List Metadata Result Set (1) 237
3.1.5.37.17 Unique Permissions Result Set (2) 238
3.1.5.37.18 NULL Unique Permissions Result Set (2) 238
3.1.5.37.19 Event Receivers Result Set (2) 238
3.1.5.37.20 List Web Parts Result Set (1) 238
3.1.5.37.21 List Metadata Result Set (2) 239
3.1.5.37.22 Unique Permissions Result Set (3) 239
3.1.5.37.23 NULL Unique Permissions Result Set (3) 239
3.1.5.37.24 Event Receivers Result Set (3) 240
3.1.5.37.25 List Web Parts Result Set (2) 240

Copyright © 2014 Microsoft Corporation.

3.1.5.37.26 Document Metadata Result Set 241
3.1.5.37.27 NULL Result Set 241
3.1.5.38 proc_GetUniqueScopesInList 241
3.1.5.38.1 Unique Permissions Result Set 242
3.1.5.38.2 NULL Unique Permissions Result Set 242
3.1.5.39 proc_GetVersion 242
3.1.5.40 proc_GetWebMetaInfo 243
3.1.5.40.1 Site Categories Result Set 244
3.1.5.40.2 Site MetaInfo Result Set 244
3.1.5.40.3 Domain Group Cache Versions Result Set 244
3.1.5.40.4 Domain Group Cache BEDS Update Result Set 244
3.1.5.40.5 Domain Group Cache WFE Update Result Set 245
3.1.5.40.6 Site Metadata Result Set 245
3.1.5.40.7 Event Receivers Result Set 245
3.1.5.41 proc_GetWebMetainfoByUrl 245
3.1.5.41.1 Site URL Result Set 246
3.1.5.41.2 Site Categories Result Set 246
3.1.5.41.3 Site Metainfo Result Set 247
3.1.5.41.4 Domain Group Cache Versions Result Set 247
3.1.5.41.5 Domain Group Cache BEDS Update Result Set 247
3.1.5.41.6 Domain Group Cache WFE Update Result Set 247
3.1.5.41.7 Site Metadata Result Set 248
3.1.5.41.8 Event Receivers Result Set 248
3.1.5.42 proc_ListDocumentVersions 248
3.1.5.42.1 Individual URL Security Result Set 249
3.1.5.42.2 NULL Individual URL Security Result Set 249
3.1.5.42.3 Document Versions Result Set 249
3.1.5.43 proc_ListUrls 250
3.1.5.43.1 Individual URL Security Result Set 251
3.1.5.43.2 NULL Individual URL Security Result Set 251
3.1.5.43.3 Server Time Result Set 251
3.1.5.43.4 Subsite List Result Set 251
3.1.5.43.5 Document Metadata Result Set 252
3.1.5.43.6 Link Info Result Set 255
3.1.5.43.7 Contained Document Metadata Result Set 255
3.1.5.44 proc_RenameUrl 259
3.1.5.44.1 Rename Result Set 261
3.1.5.44.2 Backward Link Result Set 261
3.1.5.45 proc_SecAddPrincipalToRole 261
3.1.5.45.1 Site Audit Mask Result Set 262
3.1.5.46 proc_SecAddRoleDef 262
3.1.5.46.1 Site Audit Mask Result Set 263
3.1.5.47 proc_SecAddUser 263
3.1.5.48 proc_SecAddUserToSiteGroup 266
3.1.5.49 proc_SecAddWebMembership 267
3.1.5.50 proc_SecChangeToInheritedList 267
3.1.5.50.1 Site Audit Mask Result Set 268
3.1.5.51 proc_SecChangeToInheritedWeb 268
3.1.5.51.1 Inherited Site Result Set 268
3.1.5.51.2 Site Audit Mask Result Set 269
3.1.5.52 proc_SecChangeToUniqueScope 269
3.1.5.52.1 Site Audit Mask Result Set 270
3.1.5.53 proc_SecCheckDeletedAccounts 271

Copyright © 2014 Microsoft Corporation.

3.1.5.53.1 Login Result Set 271
3.1.5.54 proc_SecCloneRoleDefinitions 271
3.1.5.54.1 Site Audit Mask Result Set 272
3.1.5.55 proc_SecCreateSiteGroup 272
3.1.5.56 proc_SecDecCurrentUsersCount 273
3.1.5.57 proc_SecGetAccountStatus 274
3.1.5.57.1 Account Status Result Set 274
3.1.5.58 proc_SecGetAclFromScope 275
3.1.5.58.1 ACL and Permission Result Set 275
3.1.5.59 proc_SecGetAllAclsForSite 275
3.1.5.59.1 Access Control List Result Set 276
3.1.5.60 proc_SecGetAllGroupsAndMembershipInfo 276
3.1.5.60.1 Groups Result Set 276
3.1.5.60.2 Group Membership Result Set 277
3.1.5.61 proc_SecGetCompleteWebRoleMemberList 278
3.1.5.61.1 Role Member Result Set 278
3.1.5.62 proc_SecGetCurrentUsersCount 278
3.1.5.62.1 User Count Result Set 279
3.1.5.63 proc_SecGetDomainGroupMapData 279
3.1.5.63.1 Domain Group Cache Versions Result Set 280
3.1.5.63.2 Domain Group Cache BEDS Update Result Set 280
3.1.5.63.3 Domain Group Cache WFE Update Result Set 280
3.1.5.64 proc_SecGetGroupById 281
3.1.5.65 proc_SecGetGroupOwner 281
3.1.5.66 proc_SecGetGroupSecurityScopes 282
3.1.5.66.1 Security Scopes Result Set 282
3.1.5.67 proc_SecGetIndividualUrlSecurityCheckEventReceivers 282
3.1.5.67.1 Individual URL Security Result Set 284
3.1.5.67.2 NULL Individual URL Security Result Set 284
3.1.5.67.3 List Metadata Result Set 284
3.1.5.67.4 Event Receivers Result Set 284
3.1.5.67.5 List Scopes Result Set 284
3.1.5.68 proc_SecGetPrincipalByEmail 285
3.1.5.68.1 Principal User Information Result Set 285
3.1.5.69 proc_SecGetPrincipalById 285
3.1.5.69.1 User Information Result Set 286
3.1.5.70 proc_SecGetPrincipalByLogin 287
3.1.5.70.1 User Information Result Set 287
3.1.5.71 proc_SecGetPrincipalByLogin20 288
3.1.5.71.1 User Information Result Set 289
3.1.5.72 proc_SecGetPrincipalDisplayInformation20 289
3.1.5.72.1 Principal Display Information Result Set 290
3.1.5.73 proc_SecGetRoleAssignments 291
3.1.5.73.1 WSSACE Result Set 292
3.1.5.74 proc_SecGetRoleBindingsForAllPrincipals 292
3.1.5.74.1 Role Assignment Result Set 292
3.1.5.75 proc_SecGetRoleDefs 292
3.1.5.75.1 Role Definition Result Set 293
3.1.5.76 proc_SecGetRoleDefsForPrincipal 294
3.1.5.76.1 Role Definition Assignment Result Set 294
3.1.5.77 proc_SecGetSecurityInfo 295
3.1.5.77.1 Security Information Result Set 295
3.1.5.78 proc_SecGetSiteAdmins 296

Copyright © 2014 Microsoft Corporation.

3.1.5.78.1 Site Administrators Result Set 296
3.1.5.79 proc_SecGetSiteGroupById 297
3.1.5.79.1 Site Group Result Set 297
3.1.5.80 proc_SecGetSiteGroupByTitle 297
3.1.5.80.1 Site Group Information Result Set 297
3.1.5.81 proc_SecGetSiteGroupByTitle20 298
3.1.5.81.1 Site Group Information Result Set 298
3.1.5.82 proc_SecGetUserAccountDirectoryPath 299
3.1.5.82.1 User Account Directory Path Result Set 299
3.1.5.83 proc_SecGetUserPermissionOnGroup 299
3.1.5.84 proc_SecListAllSiteMembers 300
3.1.5.84.1 User Information Result Set 300
3.1.5.85 proc_SecListAllWebMembers 301
3.1.5.85.1 Site Membership Result Set 301
3.1.5.86 proc_SecListGroupsInRole 302
3.1.5.86.1 Site Group Information Result Set 302
3.1.5.87 proc_SecListScopeGroups 302
3.1.5.87.1 Site Groups Result Set 303
3.1.5.88 proc_SecListScopeUsers 303
3.1.5.88.1 User Information Result Set 303
3.1.5.89 proc_SecListSiteGroupMembership 303
3.1.5.89.1 User Information Result Set 304
3.1.5.90 proc_SecListSiteGroups 304
3.1.5.90.1 Site Group Information Result Set 305
3.1.5.91 proc_SecListSiteGroupsContainingUser 305
3.1.5.91.1 Site Group Information Result Set 305
3.1.5.92 proc_SecListSiteGroupsWhichUserOwns 305
3.1.5.92.1 Site Group Information Result Set 306
3.1.5.93 proc_SecListUsersInRole 306
3.1.5.93.1 User Role Membership Result Set 307
3.1.5.94 proc_SecMigrateUser 307
3.1.5.94.1 User Migration Info Result Set 308
3.1.5.95 proc_SecReCalculateWebFGP 308
3.1.5.96 proc_SecRefreshToken 309
3.1.5.97 proc_SecRemoveGroup 309
3.1.5.97.1 DLAlias Result Set 310
3.1.5.98 proc_SecRemovePrincipalFromScope 311
3.1.5.98.1 Site Audit Mask Result Set 312
3.1.5.99 proc_SecRemoveRoleDef 312
3.1.5.99.1 Site Audit Mask Result Set 312
3.1.5.100 proc_SecRemoveUserFromScopeByLogin 312
3.1.5.100.1 Site Audit Mask Result Set 313
3.1.5.101 proc_SecRemoveUserFromSite 313
3.1.5.102 proc_SecRemoveUserFromSiteGroup 314
3.1.5.103 proc_SecRemoveUserFromSiteGroupByLogin 315
3.1.5.104 proc_SecResetItemPerm 316
3.1.5.104.1 Site Audit Mask Result Set 317
3.1.5.105 proc_SecResetWebToDefaultRoleDefinition 317
3.1.5.105.1 Site Audit Mask Result Set 319
3.1.5.106 proc_SecResolvePrincipal 319
3.1.5.106.1 Principal Information Result Set 320
3.1.5.107 proc_SecSetSiteGroupProperties 321
3.1.5.108 proc_SecSetUserAccountDirectoryPath 323

Copyright © 2014 Microsoft Corporation.

3.1.5.109 proc_SecSetWebRequestAccess 323
3.1.5.110 proc_SecUpdateAnonymousPermMask 324
3.1.5.111 proc_SecUpdateDomainGroupMapData 324
3.1.5.112 proc_SecUpdateRoleDef 325
3.1.5.112.1 Site Audit Mask Result Set 326
3.1.5.113 proc_SecUpdateUser 326
3.1.5.114 proc_SecUpdateUserActiveStatus 327
3.1.5.115 proc_TakeOverCheckOut 327
3.1.5.116 proc_UncheckoutDocument 328
3.1.5.116.1 Link Info Result Set 329
3.1.5.116.2 Document Metadata Result Set 330
3.1.5.116.3 NULL Result Set 330
3.1.5.116.4 Event Receivers Result Set 330
3.1.5.117 proc_UpdateDocBuildDependencySet 330
3.1.5.118 proc_UpdateDocument 331
3.1.5.118.1 Site List For Normalization Result Set 335
3.1.5.118.2 Site Audit Mask Result Set 335
3.1.5.118.3 Lock Information Result Set 335
3.1.5.119 proc_UpdateListItem 336
3.1.5.119.1 Item Update Result Set 344
3.1.5.120 proc_UpdateListSettings 345
3.1.5.121 proc_UpdateUserInfoInTableFromRowUpdater 349
3.1.5.122 proc_UrlToWebUrl 350
3.1.5.122.1 Web URL Result Set 351
3.1.6 Timer Events 351
3.1.7 Other Local Events 351
3.2 Web Front-End Client Details 351
3.2.1 Abstract Data Model 351
3.2.2 Timers 352
3.2.3 Initialization 352
3.2.4 Higher-Layer Triggered Events 352
3.2.5 Message Processing Events and Sequencing Rules 352
3.2.6 Timer Events 352
3.2.7 Other Local Events 352

4 Protocol Examples 353
4.1 File: GetDocsMetaInfo RPC 353
4.2 File: Open File OM 354
4.3 Group Add User To Site Group OM 356
4.4 Group Update Site Group Properties OM 359
4.5 Security: Add User to Document Library via Object Model 361
4.6 Security: Break Web Inheritance OM 363
4.7 Site Collection Lookup 364
4.7.1 Retrieving the Farm Id 364
4.7.2 Retrieving the Alternate URL Collection Ids 365
4.7.3 Retrieving the Alternate URL Collections 365
4.7.4 Alternate URL Matching 365
4.7.5 Retrieving the Web Service Ids 366
4.7.6 Retrieving the Web Application Ids 366
4.7.7 Retrieving the Web Applications 366
4.7.8 Web Application Lookup 367
4.7.9 Prefix Matching 367
4.7.9.1 Explicit Prefixes 367

Copyright © 2014 Microsoft Corporation.

4.7.9.2 Wildcard Prefixes 368
4.7.10 Site Collection Id Lookup 368
4.7.11 Building Content Database Connection String 368
4.7.11.1 Name 369
4.7.11.2 Credentials 369
4.7.11.3 Instance 369
4.7.11.4 Server Address 369
4.8 User Update User Properties OM 370
4.9 Version Negotiation 373

5 Security 374
5.1 Security Considerations for Implementers 374
5.2 Index of Security Parameters 374

6 Appendix A: Product Behavior 377

7 Change Tracking 379

8 Index 381

Copyright © 2014 Microsoft Corporation.

1 Introduction

The Windows SharePoint Services (WSS): File Operations Database Communications Protocol specifies the communication sequences used by Web Front End (WFE) and application servers to perform data query and update commands on Back End Database Servers (BEDS) as part of file, user, and group administration 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 [RFC2119]. Sections 1.5 and 1.9 are also normative but do not contain those terms. All other sections and examples in this specification are informative.

1.1 Glossary

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

access control list (ACL)
Active Directory
anonymous user
attachment
back-end database server
backward link
bot
build dependency set
change log
check out
checked out
Component Object Model (COM)
content database
content type
Coordinated Universal Time (UTC)
current user
current version
directory name
display name
displayed version
document
document identifier
document library
document stream
domain group
draft
dynamic web template
event host
event receiver
event sink
external group
feature
field
folder
form
forward link
ghosted
GUID

Copyright © 2014 Microsoft Corporation.

historical version
Hypertext Transfer Protocol (HTTP)
language code identifier (LCID)
leaf name
link
list
list identifier
list item
list item identifier
login name
metadict
minor version
page
permission level
personal view
principal
published
publishing level
Recycle Bin
result set
return code
role
role assignment
role definition
scope identifier
security principal
security provider
Security Support Provider Interface (SSPI)
server-relative URL
short-term lock
site
site collection
site collection administrator
site collection identifier
site identifier
stored procedure
store-relative form
Structured Query Language (SQL)
subsite
SystemID
thicket
thicket folder
thicket main file
thicket supporting file
Transact-Structured Query Language (T-SQL)
unghosted
user identifier
user interface (UI) version
UTF-8
version
view
web application
web bot
web discussion
Web Part

Copyright © 2014 Microsoft Corporation.

Web Part Page
Web Part zone
Welcome page
Windows code page
workflow
XML
XML namespace

The following terms are specific to this document:

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

1.2 References

References to Microsoft Open Specification documents do not include a publishing year because links are to the latest version of the 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.

[HTML] World Wide Web Consortium, "HTML 4.01 Specification", December 1999, http://www.w3.org/TR/html4/

[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 Protocol".

[MS-DTYP] Microsoft Corporation, "Windows Data Types".

[MS-FPSE] Microsoft Corporation, "FrontPage Server Extensions Remote Protocol".

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

[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

[RFC2518] Goland, Y., Whitehead, E., Faizi, A., et al., "HTTP Extensions for Distributed Authoring - WebDAV", RFC 2518, February 1999, http://www.ietf.org/rfc/rfc2518.txt

[RFC2616] Fielding, R., Gettys, J., Mogul, J., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, http://www.rfc-editor.org/rfc/rfc2616.txt

[TSQL-Ref] Microsoft Corporation, "Transact-SQL Reference (Transact-SQL)", http://msdn.microsoft.com/en-us/library/dd884419.aspx

[XPATH] Clark, J., and DeRose, S., "XML Path Language (XPath), Version 1.0", W3C Recommendation, November 1999, http://www.w3.org/TR/xpath/

Copyright © 2014 Microsoft Corporation.

1.2.2 Informative References

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

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

1.3 Overview

This protocol specifies the communication between the front-end Web server and the **back-end database server** used to satisfy requests involving file access and administration of users and groups within Windows SharePoint Services. This client-to-server protocol uses the Tabular Data Stream Protocol [MS-TDS] as its transport between the WFE, acting as a client, and the BEDS, acting as a server. The T-SQL language [TSQL-Ref] is used to define the queries and returned data which is transported over TDS.

End User Clients use remote file access protocols to communicate with the Windows SharePoint Services WFEs, specifically using FrontPage Server Extensions Remote Protocol (as specified in [MS-FPSE]), **Hypertext Transfer Protocol (HTTP)** (as specified in [RFC2616]), and WebDAV (as specified in [RFC2518]).

Further information about the interoperation of the clients with the WFE, and the WFE with the BEDS, can be found in the Windows SharePoint Services Technical Document [MS-WSSO].

1.3.1 File Operations

This protocol provides methods for retrieving and manipulating files' properties, along with support for retrieving and manipulating files' security information. When client requests for files or file information are sent to the WFE, the WFE sends a series of **stored procedure** calls to the BEDS for the requested information. The stored procedures return data that in turn can be used for further calls to other stored procedures. The WFE turns the values in the stored procedures' **return codes** and **result sets** into the data and metadata for the files requested by the client, and sends it back to the client using the same protocol used by the initial request.

1.3.2 User and Group Operations

This protocol provides methods for retrieving and manipulating information about individual users and groups, along with support for retrieving information from **Active Directory** about users. When the Object Model on the WFE operates on requests to query or update users or groups, the WFE confirms whether the data is already populated in the local objects that represent the specific user or groups. If it does not exist, the WFE sends a series of stored procedure calls to the BEDS for the requested information. The stored procedures return data, which in turn can be used for further calls to other stored procedures. The WFE turns the values in the stored procedures' return codes and result sets into objects that contain the data and metadata for the requested users or groups, and uses the objects according to implementation-specific procedures.

1.4 Relationship to Other Protocols

This protocol relies on [MS-TDS] as its transport protocol to invoke stored procedures to inspect and manipulate **document** properties via result sets and return codes. Database queries or calls to Stored Procedures, and the returned Result Sets, are written in the [TSQL-Ref] language.

Requests to a Windows SharePoint Services WFE via FrontPage Server Extensions (as described in [MS-FPSE]) and WebDAV (as described in [RFC2518]) rely on this protocol, via the WFE, to retrieve and manipulate file and security information persistently stored on the BEDS and to service requests for files and their properties from their clients.

Copyright © 2014 Microsoft Corporation.

1.5 Prerequisites/Preconditions

Unless otherwise specified, the stored procedures and any related tables MUST be present in the **content database** that is being queried on the BEDS. The tables in the content database MUST contain valid data in a consistent state in order to be queried successfully by the stored procedures.

For operations defined in this document, any file access, addition, or modification MUST be to a valid location, such as a **site** (2), **list** (1), **document library**, **folder**, or document, as defined by the data within the tables and the WFE, in order for the request to be successfully processed. The user making the request to the WFE MUST have adequate permission to access the content of the specified valid location in order for the request to be successfully processed.

1.6 Applicability Statement

This protocol is only applicable to WFE servers when communicating with the BEDS for file, user, and group administration operations.

1.7 Versioning and Capability Negotiation

The client and server in this protocol perform explicit version verifications. The client calls proc_GetVersion to retrieve version information from the server and determines whether it needs to connect to the database. The version information is stored in the Versions Table (section 2.2.7.11). This protocol supports the **Security Support Provider Interface (SSPI)** and SQL Authentication with the BEDS. These authentication methods are described in [MS-TDS].

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

Copyright © 2014 Microsoft Corporation.

2 Messages

2.1 Transport

[MS-TDS] is the transport protocol used to call the stored procedures, query **SQL** Views, or SQL Tables and return Result Codes and result sets.

2.2 Message Syntax

The following are common data types used in conjunction with this protocol. The low level data type and size are specified using commonly-known data type descriptions. It is possible that the variable is stored in multiple T-SQL data types, depending on the actual implementation of each Stored Procedure, Result Set, or Database Table. If the data type is only implemented in this protocol using a T-SQL data type, then the data type is specifically listed as a T-SQL data type.

2.2.1 Simple Data Types

The following are common simple data types used in conjunction with this protocol. When the data type is defined as being a **GUID**, as specified in [MS-DTYP] section 2.3.4, it is possible for it to be represented in T-SQL as a UniqueIdentifier or as a string. The specific T-SQL language data type used to hold the GUID is determined by the actual definition of the Stored Procedure, Result Set, or Database Table.

2.2.1.1 Content Type Identifier

A content type identifier is a numeric string value of arbitrary but limited length, which uniquely identifies a **content type**, stored on the BEDS as a **T-SQL** varbinary(512).

2.2.1.2 Document Identifier

A **document identifier** (2) is a GUID used to uniquely identify a document within a **site collection**. Specialized varieties of document identifier (2) include site identifiers (section 2.2.1.11) and list identifiers (section 2.2.1.5).

2.2.1.3 Event Receiver Identifier

An event receiver identifier is a GUID used to uniquely identify an **event receiver** within a site collection.

2.2.1.4 Feature Identifier

A feature identifier is a GUID used to uniquely identify a **feature** within a site collection.

2.2.1.5 List Identifier

A **list identifier** is a variety of document identifier (2), a GUID used to uniquely identify a list (1) within a site collection.

2.2.1.6 List Item Identifier

A **list item identifier** is a 4-byte integer value used to uniquely identify a **list item** within any list (1) in a particular site collection.

Copyright © 2014 Microsoft Corporation.

2.2.1.7 Role Identifier

A Role Identifier is a 4-byte integer value used to uniquely identify a Role Definition within a Site Collection.<1>

2.2.1.8 Scope Identifier

A **scope identifier** is a GUID used to uniquely identify a scope within a site collection.

2.2.1.9 Site Collection Identifier

A **site collection identifier** is a GUID used to uniquely identify a site collection within a Windows SharePoint Services (WSS) content database.

2.2.1.10 Site Group Identifier

A site group identifier is a 4-byte integer value used to uniquely identify a **permission level** within a site collection. Site group identifiers are assigned from the same numbering space as User Identifiers (section 2.2.1.13) and cannot overlap. Values of -1 and 0 are reserved to indicate invalid or unknown site group identifiers.<2>

2.2.1.11 Site Identifier

A **site identifier** is a variety of document identifier (section 2.2.1.2). The site identifier is a GUID used to uniquely identify a site (2) within a site collection.

2.2.1.12 SystemID

A **SystemID** is a binary value of arbitrary but limited length that uniquely identifies a **principal** (1), stored on the BEDS as a T-SQL varbinary(512).

2.2.1.13 User Identifier

A **user identifier** is a 4-byte integer value used to uniquely identify a principal (1) within a site collection. User identifiers are assigned from the same numbering space as site group identifiers and cannot overlap. Certain user identifiers are predefined such as the system account (predefined as 1073741823), the site collection owner (predefined as 1), and the secondary site collection contact (predefined as 2). Values of -1 and 0 are reserved to indicate invalid or unknown user identifiers.

2.2.1.14 View Identifier

A view identifier is a 4-byte integer value used to identify a **view** within a list (1) or document library. A view identifier is unique only within a particular list (1) or document library.

2.2.1.15 Web Part Identifier

A Web part identifier is a GUID used to uniquely identify a **Web Part** within a site collection.

2.2.1.16 Workflow Identifier

A workflow identifier is a GUID used to uniquely identify a **workflow** (2) within a site collection.

Copyright © 2014 Microsoft Corporation.

2.2.2 Bit Fields and Flag Structures

2.2.2.1 Audit Flags

Audit Flags is a 4-byte unsigned integer bit mask that tracks operations to be audited on a given object. Auditing is an implementation-specific capability of Windows SharePoint Services, which can have one or more flags set. The values of the Audit Flags bits are specified as follows.

Value	Meaning
0x00000001	Audit checkout operations.
0x00000002	Audit checkin operations.
0x00000004	Audit views.
0x00000008	Audit deletions.
0x00000010	Audit updates.
0x00000020	Audit content type updates.
0x00000040	Audit deletions of child objects.
0x00000080	Audit List Schema changes.
0x00000100	Audit security changes.
0x00000200	Audit undelete operations.
0x00000400	Audit workflow (2) operations.
0x00000800	Audit copy operations.
0x00001000	Audit move operations.
0x00002000	Audit search operations.
0xFFFFC000	Unused.

2.2.2.2 Configuration Object Status

Configuration Object Status is a 4-byte unsigned integer. The described organization is provided to facilitate the explanation of how the protocol behaves. The semantic meaning of each value is by the requirements of the service that owns the Configuration (section 2.2.6.1) Object associated with the Configuration Object Status.

The only valid values of the Configuration Object Status bits are specified in the following table. The semantic meaning of each value is determined by the implementation of the service. A possible meaning of each value is provided to facilitate the explanation of how the protocol behaves. Implementations MAY choose not to adhere to this model.

Value	Description
0x00000000	The Configuration Object is provisioned and online.
0x00000001	The Configuration Object is disabled. The resources necessary to run this Configuration Object are on the machine, but the administrator needs to provision the Configuration

Copyright © 2014 Microsoft Corporation.

Value	Description
	Object and enable it.
0x00000002	The Configuration Object is offline for some unknown reason.
0x00000003	The administrator has issued the command to unprovision the Configuration Object into a disabled state, but the unprovisioning job has not completed yet.
0x00000004	The administrator has issued the command to provision the Configuration Object and turn it online, but the provisioning job has not completed yet.
0x00000005	The administrator has issued the command to upgrade the Configuration Object into an online state, but the upgrade job has not completed yet.

2.2.2.3 Doc Flags

Doc Flags is a 4-byte unsigned integer bit mask that provides metadata about the document, which can have one or more flags set. The values of the Doc Flags bits are specified as follows.

Value	Description
0x00000001	This document contains dynamic content to 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 set if this document is an automatically generated thumbnail or web image based on another item in the store.
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 that can contain hyperlinks.
0x00000010	The document has an associated resource in the "_private" folder that is 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 **unghosted**.
0x00000080	The document is a page which contains a personalized view (showing Web Parts that are specific to the user that browsed to the page).
0x00000100	The document is a type that 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 (that is, it corresponds to an item in a non-document library list that is filtered out from file system–centric enumerations).
0x00001000	Unused.
0x00002000	The document has properties in its metainfo defining a custom order of the content types. This is only valid for folders.

Copyright © 2014 Microsoft Corporation.

Value	Description
0x00004000	The document MUST be unghosted when "undirtied" (that is, when dependency updates are performed for the document). This is used for documents such as a document library template, which is provisioned as **ghosted** but ought to be unghosted to demote content type information on the containing document library whenever that information is updated.
0x00008000	Used when a 0-byte document is saved to a document library with required check out and at least one required **field** (2). This is common in migration scenarios or with the use of older **versions** of the Windows WebDAV redirector against the Windows SharePoint Service WebDAV implementation.
0xFFFF0000	Currently unused and MUST be ignored.

2.2.2.4 Document Store Type

The Document Store Type is a 1-byte unsigned integer value that specifies the type of a document or the target of a **link** within or to a document. The only valid values for a Document Store Type are specified as follows.

Value	Description
0x00	File
0x01	folder
0x02	Site
0x80	Backward link Can only be a return value and MUST NOT be used as an input value for a stored procedure.

2.2.2.5 List Flags

List Flags is an 8-byte unsigned integer bit mask that provides metadata about the list (1), which can have one or more flags set. List Flags identify an implementation-specific capability of Windows SharePoint Services. The values of the List Flags bits are specified as follows.

Value	Meaning
0x0000000000000001	This list is an "ordered list" (for example, a Links List) and supports ordering and reordering of its items.
0x0000000000000002	This list is a "public list". This bit MUST be ignored.
0x0000000000000004	This list is "undeletable" (that is, it is crucial to the functioning of the containing site (2) or site collection).
0x0000000000000008	**Attachments** on list items are disabled. This bit MUST be set if the list is a document library or survey.
0x0000000000000010	This list is a "catalog" (for example, a Web part gallery or master page gallery).
0x0000000000000020	This list is associated with a site (2) using the meetings workspace site template and contains data scoped to each instance of a recurring meeting.
0x0000000000000040	This list MUST send alerts when a list item is assigned to a user.

Copyright © 2014 Microsoft Corporation.

Value	Meaning
0x0000000000000080	This list has versioning enabled, and supports creating **historical versions** of list items when changes occur. This bit MUST be ignored for Lists with a List Base Type of survey.
0x0000000000000100	This list MUST be hidden from enumeration functions. This is intended for lists implementing infrastructure for an application.
0x0000000000000200	This list is configured to bring up a page to fill out a **form** (2) to request access from the owner when a user is denied access while browsing its list items.
0x0000000000000400	This list has moderation enabled, requiring an approval process when content is created or modified.
0x0000000000000800	If this list is a survey, it will allow multiple responses for a given user rather than restricting users to a single response. This flag MUST be ignored for lists that do not have a List Base Type of survey.
0x0000000000001000	This list uses the value of each field's ForcedDisplay attribute when presenting data from that field (2). This is commonly used in anonymous surveys to display common placeholder text wherever the respondent's name would normally appear.
0x0000000000002000	This list MUST NOT be serialized as part of saving this site (2) as a site template.
0x0000000000004000	The List Server Template for this list can only be instantiated in the root site (2) of a given site collection.
0x0000000000008000	When a List Server Template is being created for this list, documents in the root of the list can also be serialized.
0x0000000000010000	Insertion of list items via email is enabled for this list.
0x0000000000020000	This is a "private" list. When a List Server Template based on this list is created, the new list can be given an **access control list (ACL)** so that only its owner and administrators can access the list.
0x0000000000040000	This document library requires the user to check out documents before modifying them.
0x0000000000080000	This list supports creation of **minor versions** on item revisions.
0x0000000000100000	This document library requires that users have the EditListItems right to see minor versions of documents.
0x0000000000200000	This document library requires that users have the ApproveItems right to see minor versions of documents.
0x0000000000400000	The WFE displays a user interface for manipulating multiple content types (for example, a list that contains both announcements and tasks).
0x0000000000800000	This list has had its schema customized from the version that exists in the on-disk schema file that was used to create it.
0x0000000001000000	Document parsers in this list generate thumbnail files corresponding to documents saved to this list. This bit MUST be ignored for lists that are not document libraries.
0x0000000002000000	List items in this list are visible to anyone who has access to the list itself. This is useful for shared resources such as the master page gallery, where one page can

Copyright © 2014 Microsoft Corporation.

Value	Meaning
	be used throughout a site collection in scopes with varying permissions.
0x0000000004000000	This list currently has workflows (2) associated with it.
0x0000000008000000	This list MUST NOT be automatically exported when exporting a list that references it. Exporting is an implementation-specific capability of Windows SharePoint Services.
0x0000000010000000	Applications generating server transformations of list items in this list can choose to open the list item in a browser rather than in a separate client-side application. Server transformations are performed by server-side document viewers that can allow clients to view documents without additional client software. Server transformations are an implementation-specific capability of Windows SharePoint Services.
0x0000000020000000	Creation of folders MUST be blocked in this list.
0x0000000040000000	This list disallows advanced view functionality, such as the datasheet view and views involving Web part to Web part connections.
0x0000000080000000	This list specifies custom sorting orders for the list of content types available on a per-folder basis.
0x0000000100000000	This list MUST NOT be exported as part of a migration package. Migration packages are an implementation-specific capability of Windows SharePoint Services.
0x0000000200000000	This list can have its schema cached in memory when possible, rather than retrieving the schema every time the list is accessed.
0x0000000400000000	This bit MUST NOT be returned by the database. It is used internally when manipulating the schema of sealed fields (disallowing many classes of schema modifications).
0x0000000800000000	This list MUST NOT be processed by a search crawler.
0x0000001000000000	Data from this list MUST always be included when it is saved as a List Server Template, even if not otherwise requested.
0x0000002000000000	Content type manipulation is disabled on this list.
0x0000004000000000	RSS feed syndication is disabled for this list.
0x0000008000000000	Information Rights Management (IRM) is enabled for this list.
0x0000010000000000	Expiration of IRM rights is enabled for this list. Setting this bit requires that the IRM enabled bit also be set.
0x0000020000000000	Documents that do not have a registered IRM protector will be blocked from this list.
0xFFFFFC0000000000	Unused.

2.2.2.6 Publishing Level Type

A Publishing Level Type is an 8-bit unsigned integer that describes the **publishing level** of a version of a document. Only the following values are valid for the Publishing Level Type.

Copyright © 2014 Microsoft Corporation.

Value	Description
1	The default value, referring to a **published** document shown to all users who have access to its storage location.
2	This document is in a **draft** state and is only shown to users who have permissions to see documents in a draft state.
255	This document is checked out to a particular user and is only shown to that user.

2.2.2.7 Put Flags Type

Put Flags Type is a 4-byte integer bit mask containing option flags for adding or updating a document. Zero or more of the following bit flags can be set in a Put Flags Type. The values of the **PutFlags** bits are specified as follows.

Value	Meaning
0x00000001	Update if time stamp hasn't changed.
0x00000002	Unconditionally update the document.
0x00000004	Do not add to source control.
0x00000008	Keep the document checked out.
0x00000010	Create a directory to hold the document, if necessary.
0x00000020	Check the document in.
0x00000040	Atomic: Used for multiple file update to indicate that failure to update any file causes failure.
0x00000080	Create a **thicket**.
0x00000100	Return metadata for **thicket supporting files**.
0x00000200	Permission checking has been performed.
0x00000400	Synchronize MetaInfo properties back to the document.
0x00000800	Do not increment the current internal version counter value for the containing list item. (Applies only to attachments.)
0x00001000	Create a new **displayed version** of the document, even if it is in a **short-term lock**.
0x00002000	Set only during document migration: Use client metadata for user, date and time for creation, last modification, and check in comments.
0x00004000	Treat the document as if it is in a version-enabled document library.
0x00008000	Set only for application of a theme: Do not change the value of "Modified by".
0x00010000	Publish the document.
0x00020000	Overwrite the document without updating the displayed version of the document.
0x00040000	Unpublish document: Change displayed version from published to draft.

Copyright © 2014 Microsoft Corporation.

Value	Meaning
0x00080000	Document updated from manifest file: Add document at the default publishing level.
0x00100000	The document is being added or updated as part of a system update: Do not update the last modification time and user.
0x00200000	Do not check for required fields.
0x00400000	Unused.
0x00800000	Do not increment the current internal version counter value for the document. This flag can be set only if the updater can tolerate having his or her changes overwritten by another user in the event of a conflict.
0x01000000	Unused.
0x02000000	Keep the document checked out to the user's local disk.
0xFC000000	Unused.

2.2.2.8 Rename Flags

Rename Flags is a 4-byte integer bit mask containing option flags for renaming a document. This bit mask can have zero or more flags set. The values of the **RenameFlags** bits are specified as follows.

Value	Description
0x00000000	Default behavior: Rename all dependent items.
0x00000001	Do not update all related documents.<3>
0x00000002	Create directories if they do not exist.
0x00000004	Server finds backward links in order to rename them and update the original document.
0x00000008	List thicket files or folders.
0x00000010	Fix links within the same URL subtree. Used when doing link fix up after a directory has been renamed.
0x00000020	Allow renaming of sites.
0x00000040	Allow the setting of Doc Flags bit 0x00000004 when a file's extension changes.
0x00000080	Allow update of Doc Flags bit 0x00000008 when a file's extension changes.
0x00000100	Allow renaming of sites and **directory names**.
0x00000200	Allow move into the forms (2) directory.
0x00000400	Current user can view draft documents.
0x00000800	Allow move operation on thickets with missing support files.
0xFFFFF000	Unused.

Copyright © 2014 Microsoft Corporation.

2.2.2.9 Site Collection Flags

Site Collection Flags is a 4-byte, unsigned integer bit mask that specifies properties that are global to a site collection. This bit mask can have zero or more flags set. The values of the Site Collection Flags bits are specified as follows.

Value	Meaning
0x00000001	This site collection has been write-locked, and user write operations will be blocked.
0x00000002	This site collection is fully locked, and user read and write operations will be blocked.
0x000003FC	Unused.
0x00000400	The site collection has sent a notification that disk usage is near its limit.
0x00000800	Unused.
0x00001000	The number of users in this site collection is large. Special consideration can be given to query execution plans when retrieving data.
0x00002000	The site collection has disabled syndication via RSS.
0x00004000	Notification by email is disabled for this site collection.
0x00008000	Unused.
0x00010000	Unused.
0x00020000	This site collection has been write-locked by an administrative action. Users are restricted to read-only operations.
0x00080000	The site collection is restricted to users with user accounts from a particular directory path within Active Directory.
0xFFF00000	Unused.

2.2.2.10 Site Property Flags

Site Property Flags is a 4-byte, unsigned integer bit mask that tracks property flags applied to a site (2). The site can have one or more Site Property Flags set. These flags reference implementation-specific capabilities of Windows SharePoint Services. The values of the Site Property Flags bits are specified as follows.

Value	Meaning
0x00000001	This site (2) allows display of implementation-specific user presence information in the WFE.
0x00000002	This site (2) allows display of implementation-specific enhanced user presence information in the WFE.
0x00000004	HTML views for file dialogs MUST NOT be displayed for this site (2).
0x00000008	This site has disabled syndication of list items via RSS.
0x00000010	Unused.
0x00000020	The WFE for this site (2) displays the "quick launch" navigational element.

Copyright © 2014 Microsoft Corporation.

Value	Meaning
0x00000040	The WFE for this site (2) displays a hierarchical "tree view" navigational element.
0x00000080	Document parsing is disabled for this site (2).
0x00000100	This site (2) has not yet been provisioned with a site template.
0x00000200	List schema information can be cached for lists within this site (2).
0x00000400	This site (2) has at least one uniquely secured object within it.
0x00000800	Search indexing agents can index the rendered content from ASPX pages within this site (2).
0x00001000	Search indexing agents MUST NOT index the rendered content from ASPX pages within this site (2).
0xFFFFE000	Unused.

2.2.2.11 View Flags

View Flags is a 4-byte, integer bit mask that corresponds to properties of a view. This bit mask can have zero or more flags set. The values of the View Flags bits are specified as follows.

Value	Meaning
0x00000001	Normal HTML-based view.
0x00000002	View has been modified by a client application such that it might not be compatible with the web interface for view modification. Implementations MUST restrict modifying any properties they do not understand.
0x00000004	Unused.
0x00000008	View MUST NOT be displayed in enumerations of the views of this list (1) (that is, in a view selector WFE element).
0x00000010	Value is unused and MUST be ignored.
0x00000020	View is read-only, and implementations MUST NOT allow any modifications to its properties.
0x00000040	If the query for this view returns no rows, implementations of the WFE MUST return an HTTP 410 error when displaying this view as part of an HTTP request, instead of displaying a normal empty view body.
0x00000080	Presents data in a nontabular fashion. Implementations can format results in a manner compatible with free-form presentation.
0x00000100	View suitable for displaying in an HTML-based file navigation dialog to client applications.
0x00000200	Value is unused, and implementations MUST ignore it.
0x00000400	View has functionality for aggregating data across multiple XML documents within an **XML** form library.
0x00000800	View presents a datasheet view to a rich client application.
0x00001000	Displays all items in the list recursively from the specified folder, instead of only displaying

Copyright © 2014 Microsoft Corporation.

Value	Meaning
	the immediate children of the current folder.
0x00002000	Requires that view's data be expanded based on a calendar recurrence (for example, having a view of list item data for the first Thursday of every month).
0x00004000	View is the system-created view of a user's items awaiting moderation in a moderated list.
0x00008000	System-created moderator's view of a moderated list, which displays list items pending approval.
0x00010000	Threaded view for legacy discussion boards (lists with base type 3). Implementations MUST display results in a threaded fashion, and paging of results MUST be done in terms of threads instead of by individual list items.
0x00020000	Displays HTML-based graphical charts of list item data.
0x00040000	A **personal view**, which MUST only be displayed to the user who created the view.
0x00080000	Displays data on a calendar based on date and time properties of the list items.
0x00100000	Default form of the specified type for the corresponding list.
0x00200000	Does not display any list items that are folders.
0x00400000	Displays list items based on the item order of the list (using the tp_ItemOrder column). If this list is not an ordered list, this value MUST be 0.
0x00800000	View intended for display on mobile devices.
0x01000000	View is displayed as the default view of this list when a mobile view is requested.
0x02000000	Displays historical versions of list items.
0x04000000	Displays list item data in a Gantt chart.
0x08000000	View fetches the list item for the root folder of the view, in addition to the standard behavior of fetching all list items contained within it.
0x10000000	View is the default view presented when a view is requested for a particular content type.
0x20000000	View does not display items that have not been approved. Implementations MUST refuse to show this view to **anonymous users**.
0x40000000	View MUST NOT be displayed to any user who does not possess the UseClientIntegration right.
0x80000000	Unused flag value, which MUST be ignored by client applications.

2.2.2.12 Workdays Flag

Workdays Flag is a 2-byte, bit mask that is used to specify the workdays in a week for display in a calendar. The workdays are specified as the sum or bitwise OR of the bit flags specifying each day of the week. For example, if the first day of the week is a Sunday and the workdays are Monday through Friday, the combined flags value would be 0x003E, or 62. The only valid values of the **Workdays** Flag bits are specified as follows.

Copyright © 2014 Microsoft Corporation.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5
SA	FR	TH	WE	TU	MO	SU	X	X	X	X	X	X	X	X	X

Value	Meaning
0x0001	Saturday is a workday.
0x0002	Friday is a workday.
0x0004	Thursday is a workday.
0x0008	Wednesday is a workday.
0x0010	Tuesday is a workday.
0x0020	Monday is a workday.
0x0040	Sunday is a workday.
X	MUST NOT be set.

2.2.2.13 WSS Rights Mask

WSS Rights Mask is an 8-byte, unsigned integer that specifies the rights that can be assigned to a user or site group. This bit mask can have zero or more flags set.

The values of the permission mask bits are specified as follows.

Symbolic name	Value	Description
EmptyMask	0x0000000000000000	Grant no permissions.
FullMask	0x7FFFFFFFFFFFFFFF	Grant all permissions.

The list (1) and document permissions (0x000000000000XXXX) are specified as follows.

Symbolic name	Value	Description
ViewListItems	0x0000000000000001	Allow viewing of list items in lists, documents in document libraries, and **web discussion** comments.
AddListItems	0x0000000000000002	Allow addition of list items to lists, documents to document libraries, and web discussion comments.
EditListItems	0x0000000000000004	Allow editing of list items in lists, documents in document libraries, web discussion comments, and to customize **Web Part Pages** in document libraries.
DeleteListItems	0x0000000000000008	Allow deletion of list items from lists, documents from document libraries, and web discussion comments.
ApproveItems	0x0000000000000010	Allow approval of minor versions of a list item or document.

Copyright © 2014 Microsoft Corporation.

Symbolic name	Value	Description
OpenItems	0x0000000000000020	Allow viewing the source of documents with server-side file handlers.
ViewVersions	0x0000000000000040	Allow viewing of past versions of a list item or document.
DeleteVersions	0x0000000000000080	Allow deletion of past versions of a list item or document.
CancelCheckout	0x0000000000000100	Allow discard or check in of a document that is checked out to another user.
ManagePersonalViews	0x0000000000000200	Allow creation, change, and deletion of personal views of lists.
	0x0000000000000400	Reserved.
ManageLists	0x0000000000000800	Allow creation and deletion of lists, addition or removal of fields to the schema of a list, and addition or removal of personal views of a list.
ViewFormPages	0x0000000000001000	Allow viewing of forms (2), views, and application pages, and enumerate lists.
	0x000000000000E000	Reserved.

The web level permissions (0x0000XXXXXXXX0000) are specified as follows.

Symbolic name	Value	Description
Open	0x0000000000010000	Allow access to the items contained within a site (2), list, or folder.
ViewPages	0x0000000000020000	Allow viewing of pages in a site (2).
AddAndCustomizePages	0x0000000000040000	Allow addition, modification, or deletion of HTML pages or Web Part pages, and editing of the site (2) using an editor compatible with Windows SharePoint Services.
ApplyThemeAndBorder	0x0000000000080000	Allow application of a theme or borders to the entire site (2).
ApplyStyleSheets	0x0000000000100000	Allow application of a style sheet (.css file) to the site (2).
ViewUsageData	0x0000000000200000	Allow viewing of reports on site (2) usage.
CreateSSCSite	0x0000000000400000	Allow creation of a site (2) using Self-Service Site Creation, an implementation-specific capability of Windows SharePoint Services.
ManageSubwebs	0x0000000000800000	Allow creation of a **subsite** within the site (2) or site collection.
CreateGroups	0x0000000001000000	Allow creation of a group of users that can be used anywhere within the site collection.

Copyright © 2014 Microsoft Corporation.

Symbolic name	Value	Description
ManagePermissions	0x0000000002000000	Allow creation and modification of permission levels on the site (2) and assigning permissions to users and site groups.
BrowseDirectories	0x0000000004000000	Allow enumeration of documents and folders in a site (2) using [MS-FPSE] and WebDAV interfaces.
BrowseUserInfo	0x0000000008000000	Allow viewing the information about all users of the site (2).
AddDelPrivateWebParts	0x0000000010000000	Allow addition or removal of personal Web Parts on a Web Part page.
UpdatePersonalWebParts	0x0000000020000000	Allow updating of Web Parts to display personalized information.
ManageWeb	0x0000000040000000	Allow all administration tasks for the site (2) as well as manage content.
	0x0000000F80000000	Reserved.
UseClientIntegration	0x0000001000000000	Allow use of features that launch client applications; otherwise, users MUST work on documents on their local machines and upload changes to the WFE.
UseRemoteAPIs	0x0000002000000000	Allow use of SOAP, WebDAV, or [MS-FPSE] to access the site (2).
ManageAlerts	0x0000004000000000	Allow management of alerts for all users of the site (2).
CreateAlerts	0x0000008000000000	Allow creation of email alerts.
EditMyUserInfo	0x0000010000000000	Allow users to change their own user information, such as adding a picture.
	0x0000FE0000000000	Reserved.

The Special permissions (0xXXXX000000000000) are specified as follows.

Symbolic name	Value	Description
	0x3FFF000000000000	Reserved.
EnumeratePermissions	0x4000000000000000	Allow enumeration of permissions on the site (2), list, folder, document, or list item.
	0x8000000000000000	Reserved.

2.2.3 Enumerations

2.2.3.1 Attachments Flag

The Attachments Flag is a 1-byte integer flag specifying whether an item appears to be an attachment or a folder related to attachments based on this document's URL. The following are valid values for Attachments Flag.

Copyright © 2014 Microsoft Corporation.

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 ID 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 list item attachment folder (for example, "Announcements/Attachments/17").
3	The URL is the list Attachments folder itself. The last path segment of the URL is the string "Attachments" (for example, "Announcements/Attachments").

2.2.3.2 Audit Item Type

Audit Item Type is a 2-byte integer flag that specifies the type of the object in an audit specification. The following are the only valid values for Audit Item Type.

Value	Meaning
1	A page or file in a document library.
3	A list item.
4	A list (1).
5	A folder.
6	A site (2).
7	A site collection.

2.2.3.3 Calendar Type

Calendar Type is a 2-byte integer value that specifies the type of calendar to use in a particular context. The only valid values of the Calendar Type are specified as follows.

Value	Description
0	None
1	Gregorian (localized)
3	Japanese Emperor Era
4	Taiwan Calendar
5	Korean Tangun Era
6	Hijri (Arabic Lunar)
7	Thai
8	Hebrew (Lunar)
9	Gregorian (Middle East French)

Copyright © 2014 Microsoft Corporation.

Value	Description
10	Gregorian (Arabic)
11	Gregorian (Transliterated English)
12	Gregorian (Transliterated French)
14	Korean and Japan Lunar
15	Chinese Lunar
16	Saka Era

2.2.3.4 Collation Order Enumeration

Collation Order Enumeration is a 2-byte integer value indicating collation order for textual information. The Collation Order values are mapped to Windows Collation Designator values, as defined in [Iseminger]. The only valid values of Collation Order Enumeration are specified as follows.

Value	Meaning
0	Albanian
1	Arabic
2	Chinese_PRC
3	Chinese_PRC_Stroke
4	Chinese_Taiwan_Bopomofo
5	Chinese_Taiwan_Stroke
6	Croatian
7	Cyrillic_General
8	Czech
9	Danish_Norwegian
10	Estonian
11	Finnish_Swedish
12	French
13	Georgian_Modern_Sort
14	German_PhoneBook
15	Greek
16	Hebrew
17	Hindi
18	Hungarian

Copyright © 2014 Microsoft Corporation.

Value	Meaning
19	Hungarian_Technical
20	Icelandic
21	Japanese
22	Japanese_Unicode
23	Korean_Wansung
24	Korean_Wansung_Unicode
25	Latin1_General
26	Latvian
27	Lithuanian
28	Lithuanian_Classic
29	Traditional_Spanish
30	Modern_Spanish
31	Polish
32	Romanian
33	Slovak
34	Slovenian
35	Thai
36	Turkish
37	Ukrainian
38	Vietnamese

2.2.3.5 Event Host Type

Event Host Type is a 4-byte, signed integer that specifies the type of object used as an **event host** for an event receiver. The only valid values of the Event Host Type are specified as follows.

Value	Meaning
-1	The Event Host Type is invalid.
0	The event host is a site collection.
1	The event host is a site (2).
2	The event host is a list (1).
3	The event host is a list item.
4	The event host is a content type.

Copyright © 2014 Microsoft Corporation.

Value	Meaning
5	The event host is a workflow (2).
6	The event host is a feature.

2.2.3.6 Event Receiver Type

Event Receiver Type is a 32-bit signed integer that specifies the type of an event receiver, which specifies when the handler for the event is invoked. The only valid values of the Event Receiver Type are specified as follows.

Value	Description
1	The event receiver is invoked before a list item is added.
2	The event receiver is invoked before a list item is updated.
3	The event receiver is invoked before a list item is deleted.
4	The event receiver is invoked before a list item is checked in.
5	The event receiver is invoked before a list item is checked out.
6	The event receiver is invoked before a list item checkout is reverted.
7	The event receiver is invoked before an attachment to a list item is added.
8	The event receiver is invoked before an attachment to a list item is deleted.
9	The event receiver is invoked before a document is moved.
101	The event receiver is invoked before a field (2) is added to the schema of a list (1).
102	The event receiver is invoked before a field (2) is updated in the schema of a list.
103	The event receiver is invoked before a field (2) is deleted from the schema of a list (1).
201	The event receiver is invoked before a site collection is deleted.
202	The event receiver is invoked before a site (2) is deleted.
203	The event receiver is invoked before a site (2) is moved.
10001	The event receiver is invoked after a list item is added.
10002	The event receiver is invoked after a list item is updated.
10003	The event receiver is invoked after a list item is deleted.
10004	The event receiver is invoked after a list item is checked in.
10005	The event receiver is invoked after a list item is checked out.
10006	The event receiver is invoked after a list item checkout is reverted.
10007	The event receiver is invoked after an attachment is added to a list item.
10008	The event receiver is invoked after an attachment is deleted from a list item.

Copyright © 2014 Microsoft Corporation.

Value	Description
10009	The event receiver is invoked after a document is moved.
10010	The event receiver is invoked after a document is transformed by the document transformation feature.
10101	The event receiver is invoked after a field (2) is added to the schema of a list (1).
10102	The event receiver is invoked after a field (2) is updated in the schema of a list (1).
10103	The event receiver is invoked after a field (2) is deleted from the schema of a list (1).
10201	The event receiver is invoked after a site collection is deleted.
10202	The event receiver is invoked after a site (2) is deleted.
10203	The event receiver is invoked after a site (2) is moved.
20000	The event receiver is invoked when an email message is received by a list (1).
32766	The event receiver is context sensitive and inspects the ContextType value to perform an appropriate action.
32767	The event receiver is used as part of a workflow (2).

2.2.3.7 Excluded Folder Type

Excluded Folder Type is a 4-byte integer value that indicates folders that are excluded from common listings of the subfolders in a document library due to their special **roles**. The only valid values of Excluded Folder Type are specified as follows.

Value	Meaning
0	No special handling.
1	Forms folder. This folder holds view pages within the document library or list (1).
2	Web images folder. This folder is named "_w" and holds image files in an image library.
3	Thumbnails folder. This folder is named "_t" and holds thumbnail images in an image library.
4	Root folder in a list (1) or document library.

2.2.3.8 LinkDynamic Type

LinkDynamic Type is a 1-byte value represented as a single, uppercase ASCII character that tracks various special link types. A LinkDynamic type MUST have only one value at a time. A NULL value for LinkDynamic is used for a backward link. The only valid non-NULL values of LinkDynamic are specified as follows.

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 invoke the FrontPage SmartHTML interpreter on a file.

Copyright © 2014 Microsoft Corporation.

Value	Description
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 nonabsolute link from a ghosted document that does not fall into any other category.

2.2.3.9 LinkSecurity Type

LinkSecurity Type is a 1-byte value represented as a single, uppercase ASCII character specifying the URI scheme for a link, such as HTTP or HTTPS. A LinkSecurity type MUST have only one value at a time. A NULL value for LinkSecurity is used for a backward link. The only valid non-NULL values of LinkSecurity are specified as follows.

Value	Description
H	The URL begins with "http://" (a nonsecure link using the http: scheme).
T	The URL begins with "shttp://" (an S-HTTP link using the Terisa's shttp: scheme).
S	The URL begins with "https://" (an SSL link using the https: or snews: scheme).
U	The URL is of another unknown scheme.

2.2.3.10 LinkType Types

LinkType Types is a 1-byte value represented as a single, uppercase ASCII character; it specifies type information about a link. A LinkType type MUST have only one value at a time. A NULL value for LinkType is used for a backward link. The only valid non-NULL values of a LinkType are specified as follows.

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 autogenerated 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 (2) of this list item.
K	Identical to "H", except that the link contains an HTML bookmark specifier.

Copyright © 2014 Microsoft Corporation.

Value	Description
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 Web Part connection.
P	The link is part of the markup of a Web Part 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 MasterPageFile 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**. 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 Web Part that exists in a **Web Part zone** in the containing document and is consequently not stored within the source of the containing document.

2.2.3.11 List Base Type

List Base Type is a 32-bit integer enumeration of possible base types for lists. All lists are created with one of these base types, which define implementation-specific common values for list (1) properties. The only valid values of the List Base Type are specified as follows.

Value	Meaning
0	Generic list (1)
1	Document library
3	Discussion board list (1)
4	Survey list (1)
5	Issues list (1)

2.2.3.12 List Server Template

List Server Template is a 32-bit integer enumeration of the possible values for the List Server Template defining the base structure of a list (1). The only valid values of the List Server Template are specified as follows.

Value	Meaning
-1	Invalid Template

Copyright © 2014 Microsoft Corporation.

Value	Meaning
100	Generic List Template
101	Document Library Template
102	Survey Template
103	Links Template
104	Announcements Template
105	Contacts Template
106	Events Template
107	Tasks Template
108	Discussion Template
109	Image Library Template
110	Data Sources Template
111	Web Template Catalog Template
112	User Info Catalog Template
113	Web Part Catalog Template
114	List Template Catalog Template
115	XML Form Template
116	Master Page Catalog Template
117	No Code Workflows Template
118	Workflow Process Template
119	Webpage Library Template
120	Custom Grid Template
130	Data Connection Library Template
140	Workflow History Template
150	Gantt Tasks Template
200	Meetings Template
201	Agenda Template
202	Meeting User Template
204	Decision (Meeting) Template
207	Meeting Objective Template
210	Textbox Template

Copyright © 2014 Microsoft Corporation.

Value	Meaning
211	Things To Bring (Meeting) Template
212	Homepage Library Template
301	Posts (Blog) Template
302	Comments (Blog) Template
303	Categories (Blog) Template
402	Resources List Template
403	Whereabouts List Template
404	Phone Call Memo List Template
405	Circulation List Template
420	Time Card Template
421	Holidays List Template
499	Microsoft Office IME Dictionary List Template
1100	Issue Tracking Template
1200	Administration Tasks List Template
1220	Health Rules List Template
1221	Health Reports List Template

2.2.3.13 Moderation Status

Moderation Status is a 4-byte integer indicating the moderation approval status of a list item. Configurations can require moderation approval to publish a list item or allow automatic approval. A published list item MUST have a Moderation Status of 0. The following are all possible valid values for Moderation Status.

Value	Description
0	The list item is approved.
1	The list item has been denied approval.
2	The list item is pending approval.
3	The list item is in the draft or checked out state.
4	The list item is scheduled for automatic approval at a future date.

2.2.3.14 Page Type

Page Type is a 1-byte signed integer that is used to represent the possible page types. The only valid Page Type values are specified as follows.

Copyright © 2014 Microsoft Corporation.

Value	Meaning
-1	Does not correspond to a view or a form (2) of a list (1).
0	Default view of the corresponding list (1). This view is displayed whenever this list (1) is viewed without an explicit view being specified.
1	A view of the corresponding list (1), but not the default view.
2	A view suitable for displaying in HTML file dialogs to a client application. This value is only used internally within implementation-specific code and is never stored in a database.
3	Used when dealing with views generically, regardless of their type. This value is only used internally within implementation-specific code and is never stored in a database.
4	A display form of a list (1), suitable for displaying a single list item in read-only mode.
5	Used to represent display forms of a list (1) suitable for displaying in HTML file dialogs to a client application. This value is only used internally within implementation-specific code and is never stored in a database.
6	An edit form for a list (1), suitable for presenting UI to update the properties of a list item.
7	Used to represent edit forms of a list (1) suitable for displaying in HTML file dialogs to a client application.
8	A new form for a list (1), suitable for presenting UI to create a new list item.
9	Used to represent new forms of a list (1) suitable for displaying in HTML file dialogs to a client application. This value is from a previous implementation and is no longer valid.
10	Corresponds to a template for an XML solution for a form library. This value is only used internally within implementation-specific code and is never stored in a database.

2.2.3.15 Redirect Type

Redirect Type is a 1-byte value corresponding to the type of item that a URL is redirected to. The following table contains all possible values for Redirect Type.

Value	Description
0	**Welcome page**.
1	Homepage.
2	List view – redirect to root folder.
3	Provision – redirect to a template picker page during provisioning of a new list (1).
255	None.

2.2.3.16 Role Definition Type

Role Definition Type is a 1-byte integer value that is used to represent the type of implementation-specific default and custom Role Definitions. This integer value MUST be a value enumerated in the following table.

Copyright © 2014 Microsoft Corporation.

Value	Description
0	A custom-defined role
1	Guest
2	Reader
3	Contributor
4	Web Designer
5	Administrator

2.2.3.17 Time Zone Identifier

Time Zone Identifier is a 2-byte integer value identifying a time zone. The only valid values of the Time Zone Identifier are specified as follows.

Value	Meaning
2	(GMT) Greenwich Mean Time : Dublin, Edinburgh, Lisbon, London
3	(GMT+01:00) Brussels, Copenhagen, Madrid, Paris
4	(GMT+01:00) Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna
5	(GMT+02:00) Athens, Bucharest, Istanbul
6	(GMT+01:00) Belgrade, Bratislava, Budapest, Ljubljana, Prague
7	(GMT+02:00) Minsk
8	(GMT-03:00) Brasilia
9	(GMT-04:00) Atlantic Time (Canada)
10	(GMT-05:00) Eastern Time (U.S. and Canada)
11	(GMT-06:00) Central Time (U.S. and Canada)
12	(GMT-07:00) Mountain Time (U.S. and Canada)
13	(GMT-08:00) Pacific Time (U.S. and Canada)
14	(GMT-09:00) Alaska
15	(GMT-10:00) Hawaii
16	(GMT-11:00) Midway Island, Samoa
17	(GMT+12:00) Auckland, Wellington
18	(GMT+10:00) Brisbane
19	(GMT+09:30) Adelaide
20	(GMT+09:00) Osaka, Sapporo, Tokyo
21	(GMT+08:00) Kuala Lumpur, Singapore

Copyright © 2014 Microsoft Corporation.

Value	Meaning
22	(GMT+07:00) Bangkok, Hanoi, Jakarta
23	(GMT+05:30) Chennai, Kolkata, Mumbai, New Delhi
24	(GMT+04:00) Abu Dhabi, Muscat
25	(GMT+03:30) Tehran
26	(GMT+03:00) Baghdad
27	(GMT+02:00) Jerusalem
28	(GMT-03:30) Newfoundland
29	(GMT-01:00) Azores
30	(GMT-02:00) Mid-Atlantic
31	(GMT) Casablanca, Monrovia, Reykjavik
32	(GMT-03:00) Buenos Aires, Georgetown
33	(GMT-04:00) Caracas, La Paz
34	(GMT-05:00) Indiana (East)
35	(GMT-05:00) Bogota, Lima, Quito, Rio Branco
36	(GMT-06:00) Saskatchewan
37	(GMT-06:00) Guadalajara, Mexico City, Monterrey
38	(GMT-07:00) Arizona
39	(GMT-12:00) International Date Line West
40	(GMT+12:00) Fiji Is., Kamchatka, Marshall Is.
41	(GMT+11:00) Magadan, Solomon Is., New Caledonia
42	(GMT+10:00) Hobart
43	(GMT+10:00) Guam, Port Moresby
44	(GMT+09:30) Darwin
45	(GMT+08:00) Beijing, Chongqing, Hong Kong S.A.R., Urumqi
46	(GMT+06:00) Almaty, Novosibirsk
47	(GMT+05:00) Islamabad, Karachi, Tashkent
48	(GMT+04:30) Kabul
49	(GMT+02:00) Cairo
50	(GMT+02:00) Harare, Pretoria
51	(GMT+03:00) Moscow, St. Petersburg, Volgograd

Copyright © 2014 Microsoft Corporation.

Value	Meaning
53	(GMT-01:00) Cape Verde Is.
54	(GMT+04:00) Baku
55	(GMT-06:00) Central America
56	(GMT+03:00) Nairobi
57	(GMT+01:00) Sarajevo, Skopje, Warsaw, Zagreb
58	(GMT+05:00) Ekaterinburg
59	(GMT+02:00) Helsinki, Kyiv, Riga, Sofia, Tallinn, Vilnius
60	(GMT-03:00) Greenland
61	(GMT+06:30) Yangon (Rangoon)
62	(GMT+05:45) Kathmandu
63	(GMT+08:00) Irkutsk, Ulaan Bataar
64	(GMT+07:00) Krasnoyarsk
65	(GMT-04:00) Santiago
66	(GMT+05:30) Sri Jayawardenepura
67	(GMT+13:00) Nuku'alofa
68	(GMT+10:00) Vladivostok
69	(GMT+01:00) West Central Africa
70	(GMT+09:00) Yakutsk
71	(GMT+06:00) Astana, Dhaka
72	(GMT+09:00) Seoul
73	(GMT+08:00) Perth
74	(GMT+03:00) Kuwait, Riyadh
75	(GMT+08:00) Taipei
76	(GMT+10:00) Canberra, Melbourne, Sydney
77	(GMT-07:00) Chihuahua, La Paz, Mazatlan
78	(GMT-08:00) Tijuana, Baja California
79	(GMT+02:00) Amman
80	(GMT+02:00) Beirut
81	(GMT-04:00) Manaus
82	(GMT+03:00) Tbilisi

Copyright © 2014 Microsoft Corporation.

Value	Meaning
83	(GMT+02:00) Windhoek
84	(GMT+04:00) Yerevan

2.2.3.18 Virus Status

Virus Status is a 4-byte, integer enumerated type that specifies the current virus scan status of a document. The following are valid values for Virus Status.

Value	Description
0	This document is reported as clean from viruses.
1	This document had a virus reported by the virus scanner plug-in.
2	This document had a virus reported by the virus scanner plug-in, which the scanner determines that it can remove.
3	This document had a virus previously reported, but the virus scanner determines that it successfully removed it.
4	This document had a virus reported, and the virus scanner attempted to clean it but failed.
5	This document had a virus reported, and the scanner requested that the document be deleted.
6	This document had a timeout from the virus scanner when it was last processed.

2.2.4 Binary Structures

2.2.4.1 Calendar View Options Type

Calendar View Options Type is a 1-byte value that specifies calendar options for WFE display in the following format.

0	1	2	3	4	5	6	7
FirstDayOfWeek			FirstWeekOfYear		WeekUI	Unused	

FirstDayOfWeek (3 bits): An unsigned integer specifying the first day of the week. The following are valid values for the FirstDayOfWeek value.

Bits	Description
000	Sunday
001	Monday
010	Tuesday
011	Wednesday
100	Thursday
101	Friday

Copyright © 2014 Microsoft Corporation.

Bits	Description
110	Saturday

FirstWeekOfYear (2 bits): An unsigned integer specifying how the first week of the year is handled. The following are valid values for the FirstWeekOfYear value.

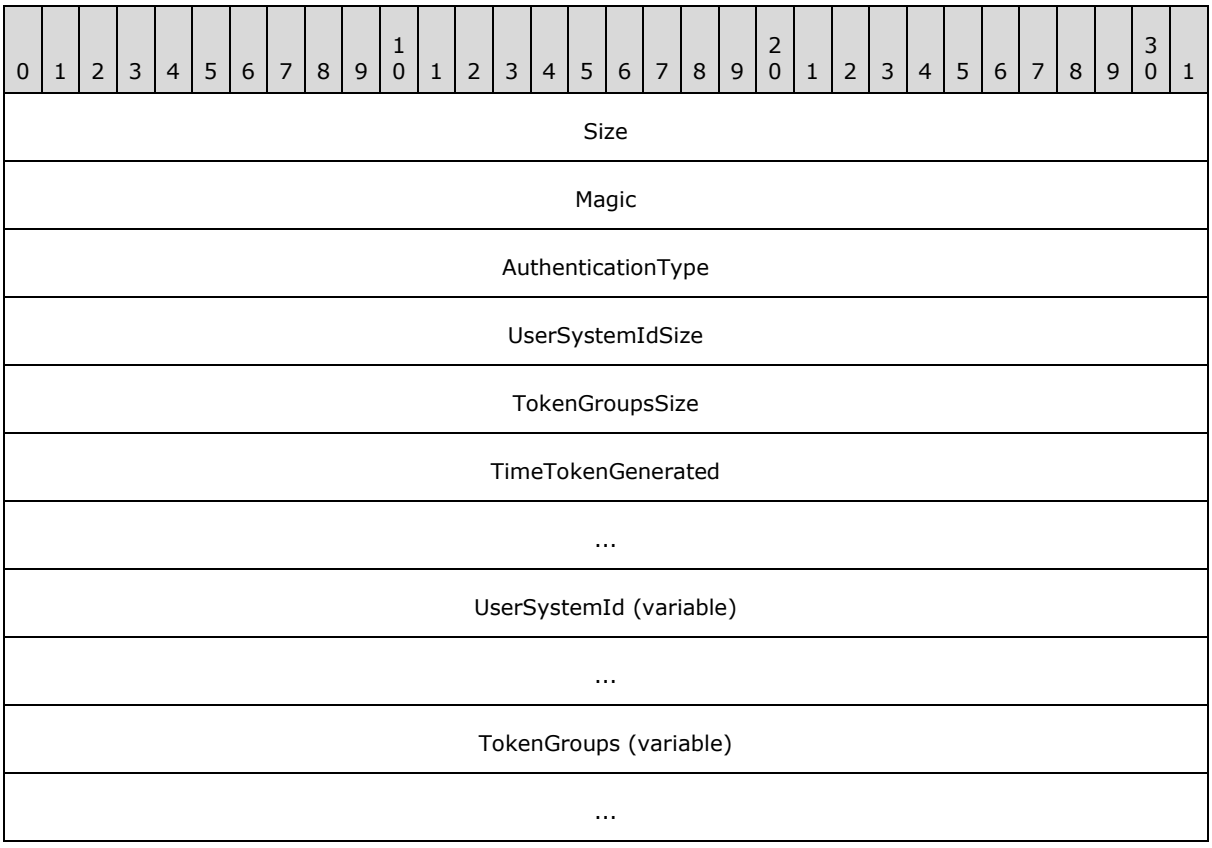
Bits	Description
00	The year starts on January 1.
01	The year starts with the first complete week.
10	The year starts with the first week of at least four days.

WeekUI (1 bit): If this bit is set, week numbers SHOULD be displayed in the WFE.

Unused (2 bits): The last 2 bits of this structure are currently unused and MUST both be set to 0.

2.2.4.2 External Group Token

The External Group Token structure is a variable-length structure associated with a principal (1) that contains a collection of the **SystemIDs** (section 2.2.1.12) for the **external groups** of which the principal is a member.



Copyright © 2014 Microsoft Corporation.

Size (4 bytes): A 4-byte, unsigned integer value specifying the length of the External Group Token in bytes.

Magic (4 bytes): A 4-byte, unsigned integer specifying the version of the External Group Token format. This version of the protocol MUST always use the value 0xCACBCECF.

AuthenticationType (4 bytes): A 4-byte, unsigned integer value specifying the authentication provider for the **SystemID** of the principal (1). The value MUST be one of the following.

Value	Meaning
0x00000001	Windows Integrated Authentication
0x00000003	ASP.NET Forms Authentication

UserSystemIdSize (4 bytes): A 4-byte, unsigned integer value specifying the length of the principal's serialized binary **SystemID** (section 2.2.1.12) in bytes.

TokenGroupsSize (4 bytes): A 4-byte, unsigned integer value specifying the length in bytes of the **TokenGroups** field (2) containing the serialized binary **SystemIDs** for the external groups.

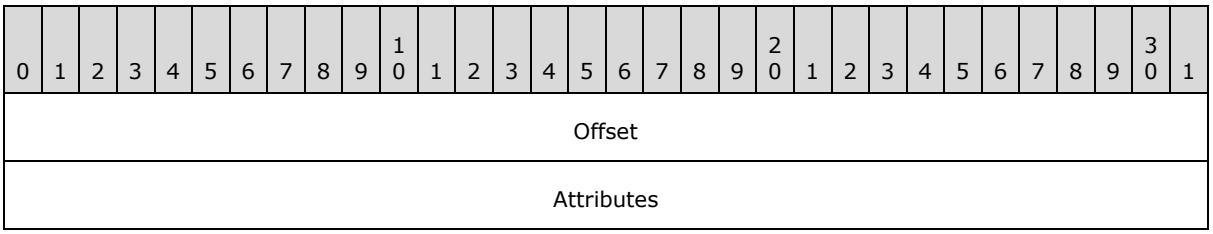
TimeTokenGenerated (8 bytes): An 8-byte, unsigned integer value specifying the time that this token was generated, stored as seconds since midnight, January 1, 1899, **Coordinated Universal Time (UTC)**.

UserSystemId (variable): A variable-length field (2) containing the serialized binary **SystemID** for the principal (1), occupying the number of bytes specified in **UserSystemIdSize**.

TokenGroups (variable): A variable-length field (2) containing a Token Groups (section 2.2.4.4) structure, which contains the serialized binary **SystemIDs** for the external groups of which the principal (1) is a member, occupying the number of bytes specified in **TokenGroupsSize**.

2.2.4.3 Token Group Offset and Attributes

The Token Group Offset and Attributes structure specifies the length and offset of the serialized binary **SystemID** (section 2.2.1.12) for a corresponding external group within a Token Groups (section 2.2.4.4) structure.



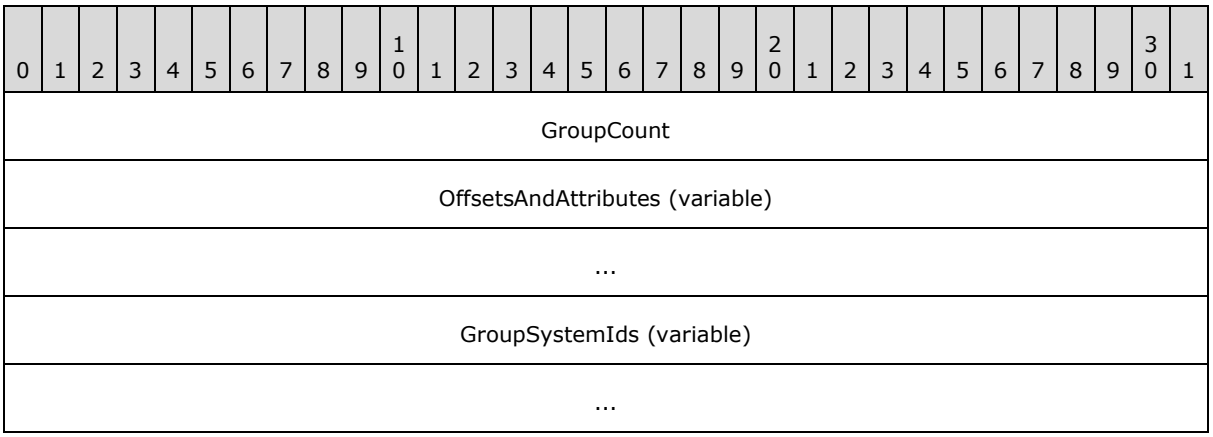
Offset (4 bytes): A 4-byte, unsigned integer value specifying the offset in bytes from the beginning of the Token Groups structure to the beginning of the serialized binary **SystemID** for this external group.

Attributes (4 bytes): A 4-byte, unsigned integer value specifying the length in bytes of the serialized binary **SystemID** for this external group.

Copyright © 2014 Microsoft Corporation.

2.2.4.4 Token Groups

The Token Groups structure contains the serialized binary **SystemIDs** (section 2.2.1.12) for a collection of external groups.



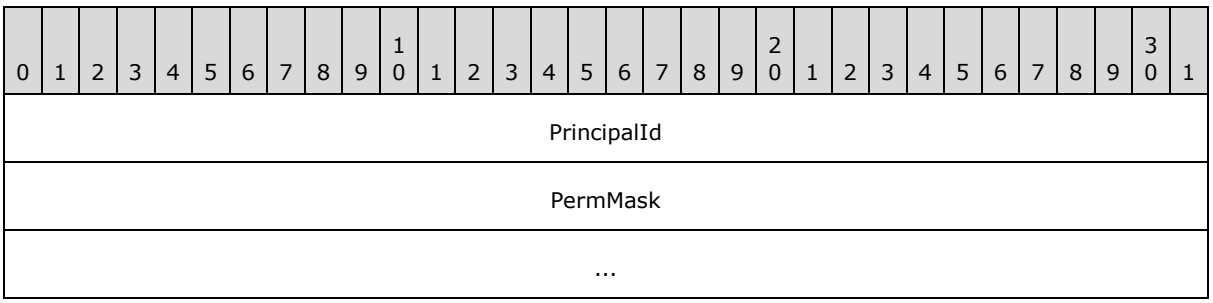
GroupCount (4 bytes): A 4-byte, unsigned integer value specifying the number of external groups' **SystemIDs** that are serialized in this structure.

OffsetsAndAttributes (variable): A variable-length serialized array containing **GroupCount** elements consisting of Token Group Offset and Attributes (section 2.2.4.3) structures specifying the lengths and offsets of the serialized binary **SystemIDs** for the corresponding external groups, one for each external group's **SystemID** in the collection.

GroupSystemIds (variable): A variable-length field (2) containing the collection of serialized binary **SystemIDs** for the external groups. Each external group's **SystemID** starts at the offset value in bytes from the beginning of the Token Groups structure and has the length in bytes specified by the Attributes value in the **OffsetsAndAttributes** array element corresponding to the external group.

2.2.4.5 WSS ACE

An ACE structure specifying the individual access rights of a principal (1).



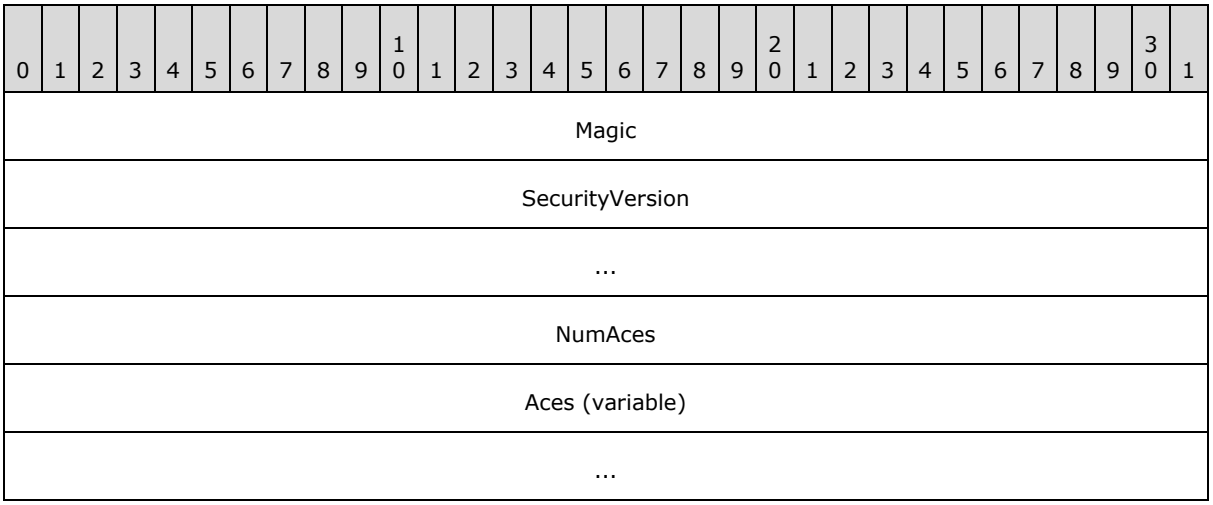
PrincipalId (4 bytes): A 4-byte, signed integer specifying the User Identifier (section 2.2.1.13) of the principal (1) for this ACE.

PermMask (8 bytes): A WSS Rights Mask (section 2.2.2.13) containing the list of rights that is granted to the principal (1).

Copyright © 2014 Microsoft Corporation.

2.2.4.6 WSS ACL Format

The WSS ACL Format structure contains an array of WSS_ACEs (section 2.2.4.5).



Magic (4 bytes): A 4-byte, unsigned integer describing the version of the ACL. This version of the protocol MUST always use the value 0xfef3.

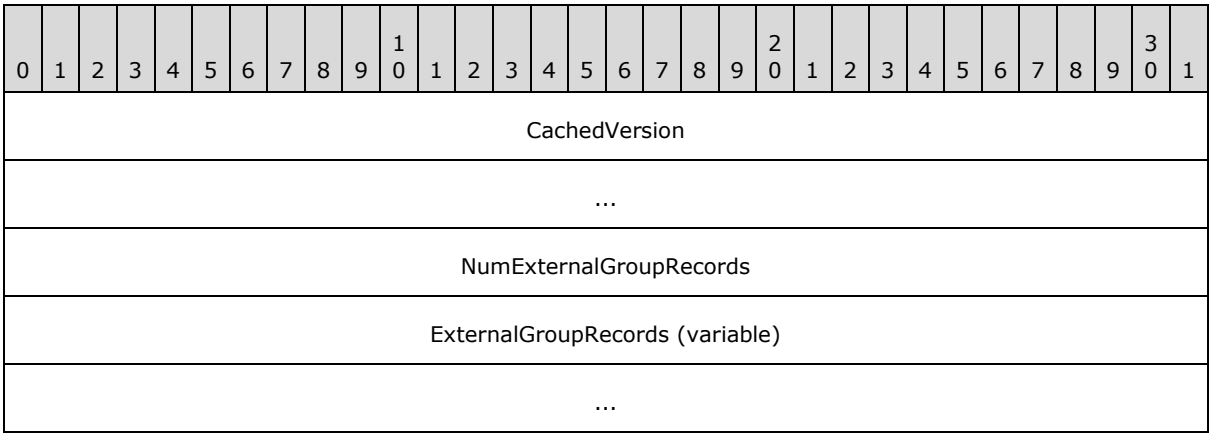
SecurityVersion (8 bytes): An 8-byte, signed integer specifying the site collection's security version value, which was used to compute the ACL. This value is not currently used.

NumAces (4 bytes): A 4-byte, unsigned integer specifying the count of WSS_ACEs within this ACL.

Aces (variable): An array of WSS_ACEs for each of the principals in this ACL.

2.2.4.7 WSS External Group Map Cache Format

The WSS External Group Map Cache Format structure contains a cache of WSS External Group Records (section 2.2.4.8) mapping external groups to the site groups that contain them.



CachedVersion (8 bytes): An 8-byte, signed integer specifying the version of this cached information.

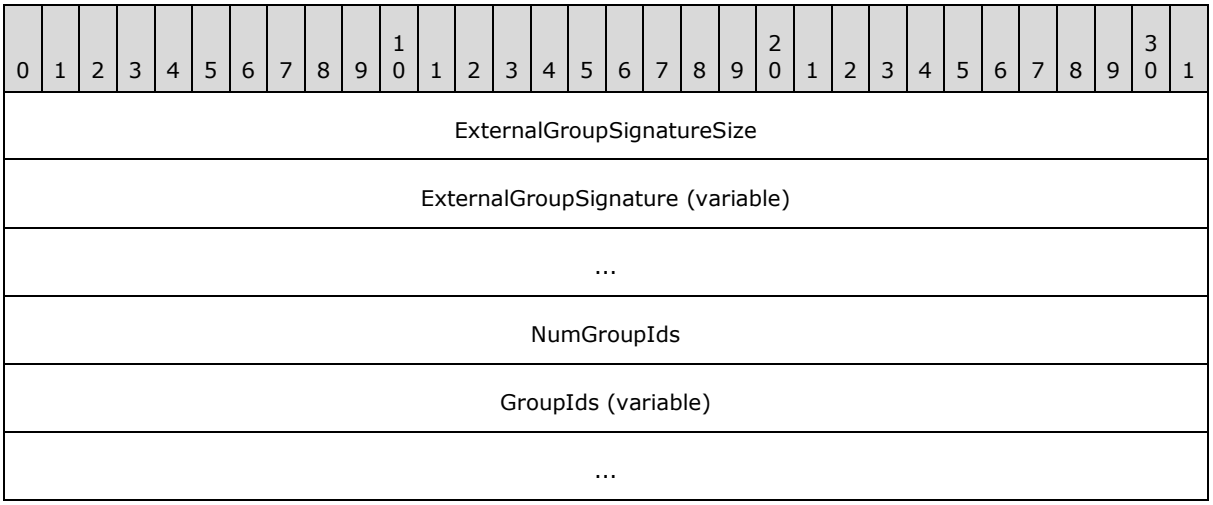
Copyright © 2014 Microsoft Corporation.

NumExternalGroupRecords (4 bytes): A 4-byte, signed integer specifying the count of WSS External Group Records present in this cache.

ExternalGroupRecords (variable): A serialized collection of WSS External Group Records.

2.2.4.8 WSS External Group Record

The WSS External Group Record structure contains the mapping between an external group and the site groups that contain it.



ExternalGroupSignatureSize (4 bytes): A 4-byte, signed integer specifying the size of the external group signature, in bytes.

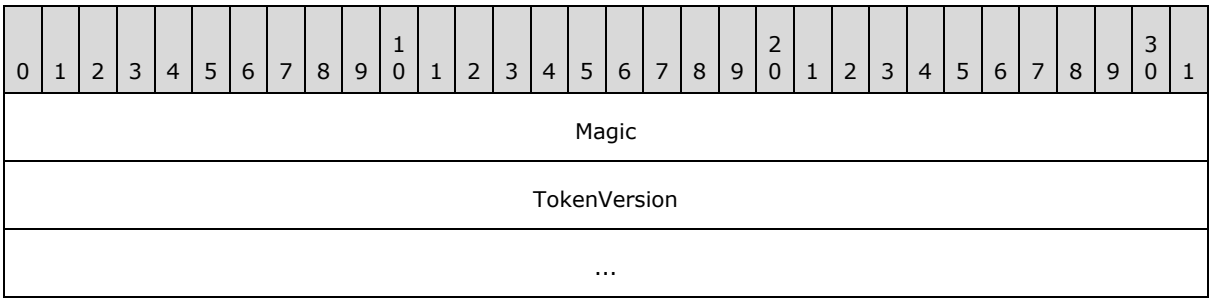
ExternalGroupSignature (variable): An array of bytes specifying the signature for the external group using **UTF-8** encoding. The number of bytes is specified by the ExternalGroupSignatureSize field (2).

NumGroupIds (4 bytes): A 4-byte, signed integer specifying the count of Site Group Identifiers (section 2.2.1.10) in this record.

GroupIds (variable): An array of 4-byte, signed integers specifying the Site Group Identifiers that contain this external group. The number of elements is specified by the NumGroupIds field (2).

2.2.4.9 WSS User Token

The WSS User Token structure contains an array of Site Group Identifiers (section 2.2.1.10).



Copyright © 2014 Microsoft Corporation.

NumGroupIds
GroupIds (variable)
...

Magic (4 bytes): A 4-byte, unsigned integer specifying the version of the token format. This version of the protocol MUST always use the value 0xdcd3.

TokenVersion (8 bytes): An 8-byte, signed integer specifying the site collection's security version value, which was used to compute the token. This value is not currently used and MUST be ignored.

NumGroupIds (4 bytes): A 4-byte, unsigned integer specifying the count of Site Group Identifiers within this token.

GroupIds (variable): An array of 4-byte integers for each of the site groups the corresponding user belongs to. The number of elements in the array is specified by the **NumGroupIds** field (2).

2.2.5 Result Sets

The following common result sets are used by this protocol.

2.2.5.1 ACL and Permission Result Set

The ACL and Permission Result Set contains information about the permissions associated with a scope in effect for a document. The ACL and Permission Result Set is defined using T-SQL syntax, as follows.

```
Acl               image,
AnonymousPermMask bigint;
```

Acl: The binary serialization of the WSS ACL Format (section 2.2.4.6) ACL for the scope in effect.

AnonymousPermMask: A WSS Rights Mask (section 2.2.2.13) indicating the rights granted to an anonymous user or a user who has no specific rights on this document.

2.2.5.2 Attachment State Result Set

The Attachment State Result Set contains the information about the attachment state of a document.

The Attachment State Result Set is defined using T-SQL syntax, as follows.

```
{IsAttachment}               bit,
{NeedManageListRight}        bit,
{Level}                      tinyint;
```

{IsAttachment}: This specifies whether the document is associated with an attachment. This value MUST be 1 if the document is an attachment, a list item attachment folder, or the list

Copyright © 2014 Microsoft Corporation.

attachment folder itself. Otherwise, this value MUST be 0. See 2.2.3.1 Attachments Flag for more information.

{NeedManagedListRight}: This specifies whether operations on this document require the user to have the ManageLists bit of the WSS Rights Mask (section 2.2.2.13) set. This value MUST be 0 if the document is not stored in a list (1).

{Level}: A Publishing Level Type (section 2.2.2.6) value specifying the publishing status of the document. This value MUST be NULL if the document does not exist.

2.2.5.3 Domain Group Cache BEDS Update Result Set

The Domain Group Cache BEDS Update Result Set contains information to be used in recomputing the **domain group** map cache, which contains the mapping of external groups to the site groups that they are members of. The presence of the Domain Group Cache BEDS Update Result Set means the database's copy of the domain group map cache is out of date and MUST be recomputed to ensure that proper security checks can be made.

The Domain Group Cache BEDS Update Result Set is defined using T-SQL syntax, as follows.

```
tp_id                           int,
tp_SystemId                     varbinary(512),
GroupId                         int;
```

tp_id: The identifier of an external group which is a member of a Site Group.

tp_SystemId: The **SystemID** (section 2.2.1.12) of the external group.

GroupId: The Site Group Identifier (section 2.2.1.10) of the Site Group containing the given domain group.

2.2.5.4 Domain Group Cache Versions Result Set

The Domain Group Cache Versions Result Set contains information about the version numbers associated with the Domain Group Map Caches on the WFE and on the BEDS for the specified Site Collection. A Domain Group Map Cache contains a serialized representation of the External Groups that are members of Site Groups in the Site Collection. The version numbers in this result set can be used to determine whether the Domain Group Map Caches on the WFE or on the BEDS are out of date and need to be refreshed. A special version number value of -2 indicates that the value MUST NOT be used for comparison.

When the Domain Group Cache Versions Result Set is returned, it MUST contain one row of version number data.

The Domain Group Cache Versions Result Set is defined using T-SQL syntax, as follows.

```
RealVersion                 bigint,
CachedVersion               bigint,
FrontEndVersion             bigint;
```

RealVersion: The most recent version number of the domain group Map Cache information.

CachedVersion: The version number of the domain group Map Cache information on the BEDS.

FrontEndVersion: The version number of the domain group Map Cache information on the WFE.

Copyright © 2014 Microsoft Corporation.

2.2.5.5 Domain Group Cache WFE Update Result Set

The Domain Group Cache WFE Update Result Set contains the binary data needed to refresh the domain group map cache. If the Domain Group Cache WFE Update Result Set is returned, it indicates that the BEDS domain group map cache is up-to-date, and the WFE cache can be refreshed if necessary.

The Domain Group Cache WFE Update Result Set is defined using T-SQL syntax, as follows.

```
DomainGroupMapCache             image;
```

DomainGroupMapCache: An image column containing the serialized domain group map cache data. If the value of FrontEndVersion is greater than or equal to the value of CachedVersion in the results in the Domain Group Cache Versions Result Set (section 2.2.5.4), this MUST be NULL.

2.2.5.6 Document Metadata Result Set

The Document Metadata Result Set returns the metadata for a document.

The Document Metadata Result Set is defined using T-SQL syntax, as follows.

```
Id                              uniqueidentifier,
{FullUrl}                       nvarchar(260),
Type                            tinyint,
MetaInfoTimeLastModified        datetime,
MetaInfo                        image,
Size                            int,
TimeCreated                     datetime,
TimeLastModified                datetime,
Version                         int,
DocFlags                        int,
{ListType}                      int,
{tp_Name}                       nvarchar(38),
{ListTitle}                     nvarchar(255),
{CacheParseId}                  uniqueidentifier,
{GhostDirName}                  int,
{GhostLeafName}                 int,
tp_Login                        nvarchar(255),
CheckoutDate                    datetime,
{CheckoutExpires}               datetime,
VirusStatus                     int,
VirusInfo                       nvarchar(255),
SetupPathVersion                tinyint,
SetupPath                       nvarchar(255),
SetupPathUser                   nvarchar(255),
NextToLastTimeModified          datetime,
UIVersion                       int,
CheckinComment                  nvarchar(1023),
WelcomePageUrl                  nvarchar(260),
WelcomePageParameters           ntext,
{tp_Flags}                      int,
Acl                             image,
AnonymousPermMask               bigint,
DraftOwnerId                    int,
Level                           tinyint,
ParentVersion                   int,
TransformerId                   uniqueidentifier,
```


Copyright © 2014 Microsoft Corporation.

```
ParentLeafName                  nvarchar(128),
ProgId                          nvarchar(255),
DoclibRowId                     int,
{tp_DefaultWorkflowId}          int,
ListId                          uniqueidentifier;
```

Id: The document identifier (section 2.2.1.2) of the requested document.

{FullUrl}: The full **store-relative form** URL for the document being requested.

Type: The Document Store Type (section 2.2.2.4) of this document.

MetaInfoTimeLastModified: A time stamp in UTC format specifying the last time the **MetaInfo** value of this document was changed, which is useful for providing minimal metadata returns to clients. This value can be NULL.

MetaInfo: A **metadict** for this document. The metadict format is specified in [MS-FPSE] section 2.2.2.2.11. This value can be NULL.

Size: The number of bytes in the document stream of the document. This value can be NULL.

TimeCreated: A time stamp in UTC format that specifies when this document was created.

TimeLastModified: A time stamp in UTC format that specifies when the document was last saved. This value does not necessarily correspond to the actual time when the document was last modified.

Version: A counter incremented any time a change is made to this document and used for internal conflict detection.

DocFlags: A Doc Flags (section 2.2.2.3) value describing the document. This value can be NULL.

{ListType}: A packed combination of the Base Type and Server Template values of the associated list (1) for this document, where the Server Template value is multiplied by 256 and added to the value of the Base Type. This value can be NULL.

{tp_Name}: The identifier of the list (1) that contains this document, formatted as a List Identifier String. This value can be NULL.

{ListTitle}: If this document is the root folder of a list (1), this contains the title of the list for display in the WFE. This value can be NULL.

{CacheParseId}: Used for concurrency detection if two different requests attempt to perform dependency update or undirtying on a document at the same time. If *@CheckCacheParseId* is set to "1", this field (2) MUST reflect the value stored for the document. Otherwise, it MUST return NULL.

{GhostDirName}: A placeholder for a directory name. This value MUST be NULL.

{GhostLeafName}: A placeholder for a **leaf name**. This value MUST be NULL.

tp_Login: If this document exists and is currently checked out, this is the **login name** of the user to whom it is checked out. In all other cases, this is NULL.

CheckoutDate: A time stamp in UTC format specifying when this document was checked out. This value can be NULL.

{CheckoutExpires}: A time stamp in UTC format that specifies when the short-term lock for this document will expire. If this date is in the past, the document SHOULD be treated as unlocked. This

Copyright © 2014 Microsoft Corporation.

value can be NULL if no short-term lock has been placed on the document. This value MUST be NULL if a user has the document checked out.

VirusStatus: An enumerated type specifying the current virus state of this document. This value can be NULL if it has not been processed by a virus scanner. Valid values are listed in the Virus Status (section 2.2.3.18) section.

VirusInfo: A string containing a provider specific message returned by the virus scanner when it last processed the document. This value can be NULL if it has not been processed by a virus scanner.

SetupPathVersion: For a ghosted document, this governs the setup path location that the **SetupPath** fragment is relative to. This value MUST NOT be NULL. The following values are valid.

Value	Description
2	The SetupPath is relative to the install location of Windows SharePoint Services 2.0 on the WFE (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\60)
3	The SetupPath is relative to the install location of Windows SharePoint Services 3.0 on the WFE (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\12).

SetupPath: For a document that is now or once was ghosted, this contains the setup path fragment relative to the base setup path described earlier by the **SetupPathVersion** value, where the content stream of this document can be found. This value can be NULL.

SetupPathUser: If this document is now or once was ghosted, this contains the login name of the user that created the ghosted document. This value can be NULL.

NextToLastTimeModified: The value of **TimeLastModified** from the previous time when the document was last saved. If the **NextToLastTimeModified** value matches the **TimeLastModified** value when the rename occurred, and the client has a document that it has successfully fixed up, the client can safely submit the document to the server despite what appears to be an intervening edit to the document. This value can be NULL.

UIVersion: The **UI version** number for the document. This value MUST be NULL in the case of a document that does not exist.

CheckinComment: An optional description provided when a document is checked in or published, which could be displayed in version management UI. This value can be NULL.

WelcomePageUrl: If this document is a folder, this specifies an optional page to redirect to when the folder is requested with an HTTP GET operation. The URL is relative to the URL of the folder itself and MUST be subsumed by that folder. Attempts to break out of the folder such as "../../somepage.aspx" are not valid. This value can be NULL.

WelcomePageParameters: Contains optional "search" parameters to specify as part of the **WelcomePageUrl** value. These parameters start at either the query string signifier '?' or the bookmark signifier '#'. This value can be NULL.

{tp_Flags}: A List Flags (section 2.2.2.5) value describing the list (1) that contains this document. This value can be NULL.

Acl: The binary serialization of the WSS ACL Format (section 2.2.4.6) ACL for this document. This is either explicitly defined or inherited from the parent object of the document. This value can be NULL.

Copyright © 2014 Microsoft Corporation.

AnonymousPermMask: A WSS Rights Mask (section 2.2.2.13) that indicates the rights granted to a user that is anonymous, or has no specific rights, to this document. This value can be NULL.

DraftOwnerId: The User Identifier (section 2.2.1.13) of the user that published this document as a draft. This value MUST only be non-NULL if the requested document exists and is a draft version.

Level: A Publishing Level Type (section 2.2.2.6) value specifying the publishing status of this document.

ParentVersion: If the document is a transformed version of another document, this is the UI version value from the original document. This value MUST be NULL if the document is not the product of a document transformation.

TransformerId: If the document is a transformed version of another document, this is the GUID of the agent that performed the transformation. This value MUST be NULL if the document is not the product of a document transformation.

ParentLeafName: If the document is a transformed version of another document, this is the leaf name of the original document. The original document MUST be in the same folder as the transformed document. If either is moved, the relationship is broken. This value MUST be NULL if the document is not the product of a document transformation.

ProgId: Designates a preferred application to open the document. The ProgId is used to distinguish between different applications that save files with a given file extension (that is, different editors for HTML or XML files). This value MUST be NULL if the parser did not specify a ProgId when the document was saved.

DoclibRowId: The document library row identifier for this document. If the requested document is not contained in a list (1), this value MUST be NULL.

{tp_DefaultWorkflowId}: The Workflow Identifier (section 2.2.1.16) corresponding to the workflow (2) to be invoked if this document is in a moderated list (1) and this document is submitted for approval as part of a check in. This value MUST be NULL.

ListId: The List Identifier (section 2.2.1.5) of the list (1) that contains the requested document. If the document is not contained in a list (1), this value MUST be NULL.

2.2.5.7 Document Version Metadata Result Set

The Document Version Metadata Result Set contains the metadata about the requested version of the document.

The Document Version Metadata Result Set MUST only be returned if *@Version* is not negative. If the document exists, the Document Version Metadata Result Set MUST contain one row; otherwise, it MUST contain no rows.

The Document Version Metadata Result Set is defined using T-SQL syntax, as follows.

```
Id                          uniqueidentifier,
{FullUrl}                   nvarchar(260),
Type                        tinyint,
MetaInfoTimeLastModified    datetime,
MetaInfo                    image,
Size                        int,
TimeCreated                 datetime,
TimeCreated                 datetime,
Version                     int,
```


Copyright © 2014 Microsoft Corporation.

```
DocFlags                    int,
{ListType}                  int,
{tp_Name}                   nvarchar(38),
{ListTitle}                 nvarchar(255),
{CacheParseId}              uniqueidentifier,
{GhostDirName}              nvarchar(256),
{GhostLeafName}             nvarchar(128),
tp_Login                    nvarchar(255),
CheckoutDate                datetime,
{CheckoutExpires}           datetime,
VirusStatus                 int,
VirusInfo                   nvarchar(255),
SetupPathVersion            tinyint,
SetupPath                   nvarchar(255),
SetupPathUser               nvarchar(255),
NextToLastTimeModified      datetime,
Version                     int,
CheckinComment              nvarchar(1023),
WelcomePageUrl              nvarchar(260),
WelcomePageParameters       ntext,
{tp_Flags}                  bigint,
Acl                         image,
AnonymousPermMask           int,
DraftOwnerId                int,
Level                       tinyint,
ParentVersion               int,
TransformerId               uniqueidentifier,
ParentLeafName              nvarchar(128),
ProgId                      nvarchar(255),
DoclibRowId                 int,
{tp_DefaultWorkflowId}      uniqueidentifier,
ListId                      uniqueidentifier;
```

Id: The document identifier (section 2.2.1.2) of the requested document.

{FullUrl}: The full store-relative form URL for the document being requested.

Type: The Document Store Type (section 2.2.2.4) of this document.

MetaInfoTimeLastModified: A time stamp in UTC format specifying the last time the **Metainfo** value of this document was changed, which is useful for providing minimal metadata returns to clients. This value can be NULL.

MetaInfo: A metadict for this document version. The metadict format is specified in [MS-FPSE] section 2.2.2.2.11. This value can be NULL.

Size: The number of bytes in the document stream of the document version requested. This value can be NULL.

TimeCreated: A time stamp in UTC format specifying when this document was created.

TimeCreated: A time stamp in UTC format specifying when the document was last saved. This corresponds to the TimeCreated or **TimeLastModified** of the document version requested.

Version: A counter incremented any time a change is made to this document and used for internal conflict detection.

DocFlags: A Doc Flags (section 2.2.2.3) value describing this document. This value can be NULL.

Copyright © 2014 Microsoft Corporation.

{ListType}: A packed combination of the Base Type and Server Template values of the associated list (1) for this document, where the Server Template value is multiplied by 256 and added to the value of the Base Type. This value MUST be NULL.

{tp_Name}: The identifier of the list (1) that contains this document, formatted as a List Identifier (section 2.2.1.5). This value MUST be NULL.

{ListTitle}: If this document is the root folder of a list (1), this contains the title of the list (1) for display in the user interface. This value MUST be NULL.

{CacheParseId}: Used for concurrency detection if two different requests attempt to perform dependency update or undirtying on a document at the same time. If *@CheckCacheParseId* is set to "1", this field (2) MUST reflect the value stored for the document. Otherwise, it MUST return NULL.

{GhostDirName}: A placeholder for a directory name. This value MUST be NULL.

{GhostLeafName}: A placeholder for a leaf name. This value MUST be NULL.

tp_Login: If this document exists and is currently checked out, this is the login name of the user to whom it is checked out. In all other cases, this is NULL.

CheckoutDate: A time stamp in UTC format specifying when this document was checked out. This value can be NULL.

{CheckoutExpires}: A time stamp in UTC format specifying when the short-term lock for this document will expire. If this date is in the past, the document SHOULD be treated as unlocked. This value can be NULL if no short-term lock has been placed on the document. This value MUST be NULL if a user has the document checked out.

VirusStatus: An enumerated type specifying the current virus state of this document version. This value can be NULL if it has not been processed by a virus scanner. Valid values are listed in the Virus Status (section 2.2.3.18) section.

VirusInfo: A string containing a provider specific message returned by the virus scanner when it last processed the document. This value can be NULL if it has not been processed by a virus scanner.

SetupPathVersion: For a ghosted document, this governs the setup path location that the **SetupPath** fragment is relative to. This value MUST NOT be NULL. The following are values are valid.

Value	Description
2	The **SetupPath** is relative to the install location of Windows SharePoint Services 2.0 on the WFE (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\60)
3	The **SetupPath** is relative to the install location of Windows SharePoint Services 3.0 on the WFE (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\12).

SetupPath: For a document that is now or once was ghosted, this contains the setup path fragment relative to the base setup path described earlier by the **SetupPathVersion** value, where the content stream of this document can be found. This value can be NULL.

SetupPathUser: If this document is now or once was ghosted, this contains the login name of the user that created the ghosted document. This value can be NULL.

Copyright © 2014 Microsoft Corporation.

NextToLastTimeModified: The value of **TimeLastModified** from the previous time when the document was last saved. If the **NextToLastTimeModified** value matches the **TimeLastModified** value when the rename occurred and the client has a document that it has successfully fixed up, the client can safely submit the document to the server despite what appears to be an intervening edit to the document. This value can be NULL.

Version: The UI version number for the document. This value can be produced as the integer counter described earlier (the first Version field (2)). This value MUST match **@***Version*.

CheckinComment: An optional description provided when a document is checked in, which could be displayed in version management UI. This value can be NULL.

WelcomePageUrl: If this document is a folder, this specifies an optional page to redirect to when the folder is requested with an HTTP GET operation. The URL is relative to the URL of the folder itself and MUST be subsumed by that folder. Attempts to break out of the folder such as "../../somepage.aspx" are not valid. This value can be NULL.

WelcomePageParameters: Contains optional "search" parameters to specify as part of the **WelcomePageUrl** value. These parameters start at either the query string signifier '?' or the bookmark signifier '#'. This value can be NULL.

{tp_Flags}: A List Flags (section 2.2.2.5) value describing the list (1) that contains this document. This value MUST be NULL.

Acl: The binary serialization of the WSS ACL Format (section 2.2.4.6) ACL for this document. This is either explicitly defined or inherited from the parent object of the document. This value can be NULL.

AnonymousPermMask: A WSS Rights Mask (section 2.2.2.13) that indicates the rights granted to a user that is anonymous, or has no specific rights, to this document. This value can be NULL.

DraftOwnerId: The User Identifier (section 2.2.1.13) of the user that published this document as a draft. This value is only non-NULL if the requested document is a draft version.

Level: A Publishing Level Type (section 2.2.2.6) value specifying the publishing status of this document. This value MUST match **@**Level.

ParentVersion: If the document is a transformed version of another document (see **Doc Flags** value 0x00000400, section 2.2.2.3), this is the UIVersion value from the parent document. This value MUST be NULL if the document is not a child of a document transformation.

TransformerId: If the document is a transformed version of another document, this is the GUID of the agent that performed the transformation. This value MUST be NULL if the document is not the product of a document transformation.

ParentLeafName: If the document is a transformed version of another document, this is the leaf name of the original document. The original document MUST be in the same folder as the transformed document. If either is moved, the relationship is broken. This value MUST be NULL if the document is not the product of a document transformation.

ProgId: Designates a preferred application to open the document. The **ProgId** is used to distinguish between different applications that save files with a given file extension (that is, different editors for HTML or XML files). This value MUST be NULL if the parser did not specify a **ProgId** when the document was saved.

DoclibRowId: The document library row identifier for this document. If the requested document is not contained in a list (1), this value MUST be NULL.

Copyright © 2014 Microsoft Corporation.

{tp_DefaultWorkflowId}: The Workflow Identifier (section 2.2.1.16) corresponding to the workflow (2) invoked if this document is in a moderated list (1) and this document is submitted for approval as part of a check in. This value MUST be NULL.

ListId: The List Identifier of the list (1) containing the requested document. If the document is not contained in a list (1), this value MUST be NULL.

2.2.5.8 Empty Result Set

The Empty Result Set is defined using T-SQL syntax, as follows.

```
{No column name}     Not Applicable
```

2.2.5.9 Event Receivers Result Set

The Event Receivers Result Set contains information about the event receivers defined for an Event Host.

The Event Receivers Result Set MUST contain one row for each event receiver registered for the Event Host.

The Event Receivers Result Set is defined using T-SQL syntax, as follows.

```
Id                          uniqueidentifier,
Name                        nvarchar(256),
SiteId                      uniqueidentifier,
WebId                       uniqueidentifier,
HostId                      uniqueidentifier,
HostType                    int,
ItemId                      int,
DirName                     nvarchar(256),
LeafName                    nvarchar(128),
Type                        int,
SequenceNumber              int,
Assembly                    nvarchar(256),
Class                       nvarchar(256),
Data                        nvarchar(256),
Filter                      nvarchar(256),
SourceId                    varbinary(512),
SourceType                  int,
Credential                  int,
ContextType                 varbinary(16),
ContextEventType            varbinary(16),
ContextId                   varbinary(16),
ContextObjectId             varbinary(16),
ContextCollectionId         varbinary(16);
```

Id: The Event Receiver Identifier of the event receiver.

Name: The **display name** of the event receiver.

SiteId: The Site Collection Identifier (section 2.2.1.9) for a site collection that contains the event host on which the event receiver is registered.

Copyright © 2014 Microsoft Corporation.

WebId: The Site Identifier (section 2.2.1.11) for the site (2) that contains the event host on which the event receiver is registered.

HostId: The document identifier (section 2.2.1.2) for the event host on which the event receiver is registered.

HostType: The Event Host Type (section 2.2.3.5) of the event host on which the event receiver is registered.

ItemId: Reserved. The server MUST default this value to NULL, and a client MUST NOT use this value.

DirName: Reserved. The server MUST default this value to NULL, and a client MUST NOT use this value.

LeafName: Reserved. The server MUST default this value to NULL, and a client MUST NOT use this value.

Type: The Event Receiver Type (section 2.2.3.6) of the event receiver.

SequenceNumber: An ordinal value that determines the relative order in which the event receiver is triggered. **SequenceNumber** MUST be greater than or equal to 0 and less than or equal to 65535.

Assembly: The name of the .NET assembly that contains the implementation of the event receiver.

Class: The name of the .NET class definition that contains the implementation of the event receiver.

Data: Additional data persisted on behalf of the event receiver implementation, to be passed to the event receiver.

Filter: Reserved. The server MUST default this value to NULL, and a client MUST NOT use this value.

SourceId: The feature identifier or content type identifier of the feature or content type that registered the event receiver. If the event receiver was not registered by a feature or content type, the value MUST be NULL.

SourceType: An integer value that specifies the source of the registration for the event receiver. The **SourceType** value MUST be one of the following values.

Value	Description
0	The event receiver does not come from a specially treated source.
1	A content type registered the event receiver.
2	A feature registered the event receiver.

Credential: Reserved. The server MUST default this value to NULL, and a client MUST NOT use this value.

ContextType: The workflow identifier of the workflow (2) that the event relates to, if any.

ContextEventType: Reserved. The server MUST default this value to NULL, and a client MUST NOT use this value.

ContextId: The workflow identifier for the registered workflow (2), if any.

Copyright © 2014 Microsoft Corporation.

ContextObjectId: The document identifier (2) for the SharePoint Object instance that the Workflow (2) is registered against, if any.

ContextCollectionId: A workflow identifier for the workflow instance that manages the event receiver, if any.

2.2.5.10 Individual URL Security Result Set

The Individual URL Security Result Set contains security information about a document.

The Individual URL Security Result Set is defined using T-SQL syntax, as follows.

```
{ListId}                    uniqueidentifier,
Acl                         image,
AnonymousPermMask           bigint,
{IsAttachment}              bit,
{NeedManageListRight}       bit,
{BaseType}                  int,
{ExcludedType}              int,
{ListFlags}                 bigint,
{Level}                     tinyint,
{DraftOwnerId}              int,
{DoclibRowId}               int;
```

{ListID}: The List Identifier (section 2.2.1.5) of the list (1) or document library containing the document location.

Acl: The WSS ACL (section 2.2.4.6) for the scope associated with the document location.

AnonymousPermMask: The WSS Rights Mask (section 2.2.2.13) that applies to an anonymous user, or a user with no assigned rights, in the scope associated with the document location.

{IsAttachment}: A bit flag specifying whether the document location is an attachment or an attachment folder. This value MUST be set to 1 if the document location is an attachment or attachment folder; otherwise, it MUST be 0.

{NeedManagedListRight}: A bit flag specifying whether the **current user** has the **ManageLists** bit of the WSS Rights Mask set to write to the document location. This value MUST be set to 1 if the current user MUST have the **ManageLists** bit of the WSS Rights Mask set in the scope of the document location in order to write the document; otherwise, it MUST be 0.

{BaseType}: The List Base Type section (section 2.2.3.11) of the list (1) or document library containing the document location.

{ExcludedType}: Contains an enumerated value specifying whether the document location is within a special folder type in the containing list (1) or document library. The following values are valid.

Value	Description
0	The document location is not contained in a special folder.
1	The document location is, or is contained within, a forms folder: a folder named "Forms" within a document library.
2	The document location is, or is contained within, a web image folder: a folder named "_w" within

Copyright © 2014 Microsoft Corporation.

Value	Description
	a document library.
3	The document location is, or is contained within, a thumbnail folder: a folder named "_t" within a document library.
4	The document location is at the root folder of the list (1) or document library.

{ListFlags}: The List Flags (section 2.2.2.5) of the list (1) or document library containing the document location.

{Level}: The Publishing Level Type (section 2.2.2.6) value specified for the document, or if the parameter was NULL, the Publishing Level Type of the **current version** of the document for the current user.

{DraftOwnerId}: The User Identifier (section 2.2.1.13) of the principal (1) that published this document as a draft. This value MUST be NULL if the document does not exist or does not have a draft version.

{DoclibRowId}: The row identifier of the document in the containing list (1) or document library.

2.2.5.11 Link Information Result Set

The Link Information Result Set returns information about each **forward link** from the document and backward link to the document within the site collection.

The Link Information Result Set is defined using T-SQL syntax, as follows.

```
LinkDirName                 nvarchar(256),
LinkLeafName                nvarchar(128),
LinkType                    tinyint,
LinkSecurity                tinyint,
LinkDynamic                 tinyint,
LinkServerRel               bit,
LinkStatus                  tinyint,
PointsToDir                 bit,
WebPartId                   uniqueidentifier,
LinkNumber                  int,
WebId                       uniqueidentifier,
Search                      ntext,
FieldId                     uniqueidentifier;
```

LinkDirName: Contains the directory name of the link target.

LinkLeafName: Contains the leaf name of the link target.

LinkType: The **LinkType** (section 2.2.3.10) value of the link. This value MUST be NULL for a backward link.

LinkSecurity: The **LinkSecurity** (section 2.2.3.9) value of the forward link, which specifies whether the scheme of the link is HTTP or HTTPS. This value MUST be NULL for a backward link.

LinkDynamic: The **LinkDynamic** (section 2.2.3.8) value of the forward link, which specifies whether the link is to a special type of target. This value MUST be NULL for a backward link.

Copyright © 2014 Microsoft Corporation.

LinkServerRel: A bit flag which specifies whether or not the link is a **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 (section 2.2.2.4) value of the document targeted by a forward link. This value MUST be 128 for a backward link. If the forward link target is a document that does not exist, or if this 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 link target is a directory where a welcome page is specified, the link MUST be changed to the URL of the welcome page and **PointsToDir** MUST be set to 1 so that the link can be distinguished from an explicit link to the welcome page; otherwise this value MUST be 0.

WebPartId: This value MUST be NULL.

LinkNumber: This value MUST be NULL.

WebId: This value MUST be NULL.

Search: This value MUST be NULL.

FieldId: This value MUST be NULL.

2.2.5.12 List Metadata Result Set

The List Metadata Result Set contains the metadata associated with a list (1).

The List Metadata Result Set is defined using T-SQL syntax, as follows.

```
tp_Id                           uniqueidentifier,
tp_Title                        nvarchar(255),
tp_Modified                     datetime,
tp_Created                      datetime,
tp_LastDeleted                  datetime,
tp_Version                      int,
tp_BaseType                     int,
tp_FeatureId                    uniqueidentifier,
tp_ServerTemplate               int,
DirName                         nvarchar(256),
LeafName                        nvarchar(128),
DirName                         nvarchar(256),
LeafName                        nvarchar(128),
tp_ReadSecurity                 int,
tp_WriteSecurity                int,
tp_Description                  ntext,
{tp_Fields}                     ntext,
tp_Direction                    int,
AnonymousPermMask               bigint,
tp_Flags                        bigint,
tp_ThumbnailSize                int,
tp_WebImageWidth                int,
tp_WebImageHeight               int,
tp_ImageUrl                     nvarchar(255),
tp_ItemCount                    int,
tp_Author                       int,
tp_HasInternalFGP               bit,
tp_ScopeId                      uniqueidentifier,
```


Copyright © 2014 Microsoft Corporation.

```
Acl                                 image,
tp_EventSinkAssembly                nvarchar(255),
tp_EventSinkClass                   nvarchar(255),
tp_EventSinkData                    nvarchar(255),
tp_EmailInsertsFolder               nvarchar(255),
tp_EmailInsertsLastSyncTime         nvarchar(50),
tp_EmailAlias                       nvarchar(128),
tp_WebFullUrl                       nvarchar(256),
tp_WebId                            uniqueidentifier,
tp_WebTitle                         nvarchar(255),
tp_Web                              int,
tp_WebLanguage                      int,
tp_WebCollation                     smallint,
tp_SendToLocation                   nvarchar(512),
{tp_MaxMajorVersionCount}           int,
{tp_MaxMajorwithMinorVersionCount}  int,
tp_MaxRowOrdinal                    int,
tp_ListDataDirty                    int,
tp_DefaultWorkflowId                uniqueidentifier,
tp_ContentTypes                     ntext,
tp_Subscribed                       bit;
```

tp_Id: The List Identifier (section 2.2.1.5) of the list (1).

tp_Title: The title of this list (1) for display in the WFE.

tp_Modified: A time stamp in UTC format specifying when this list (1) was last modified.

tp_Created: A time stamp in UTC format specifying when this list (1) was created.

tp_LastDeleted: A time stamp in UTC format specifying when an item was last deleted from this list (1). This value MUST default to NULL if no item has been deleted.

tp_Version: A counter incremented any time a change is made to the schema or other properties of this list (1) and used for internal conflict detection.

tp_BaseType: The List Base type (section 2.2.3.11) of this list (1). The following values are valid.

Value	Description
0	"Generic" list (1)
1	Document library
3	"Discussion board" list (1)
4	"Survey" list (1)
5	"Issues" list (1)

tp_FeatureId: The Feature Identifier (section 2.2.1.4) for the feature that defines the base schema of this list (1).

tp_ServerTemplate: The identifier for the template included in the feature definition or site definition that defines the base structure of this list (1).

Copyright © 2014 Microsoft Corporation.

DirName: The directory name of the location that contains this list (1).

LeafName: The leaf name of the location that contains this list (1).

DirName: The directory name of the default template document in the list (1). This value can be NULL if a template document is not defined for this list (1).

LeafName: The leaf name of the default template document in the list (1). This value can be NULL if a template document is not defined for this list (1).

tp_ReadSecurity: This signifies special restrictions that can be placed on list item access. The following values are valid.

Value	Description
1	No special restrictions.
2	Users can see only their own list items. The WFE MUST NOT display list items to users without the ManageLists right unless the list item was created by that user (for example, **tp_**Author = **@**UserId).

tp_WriteSecurity: This signifies special restrictions that can be placed on list item update. The following values are valid.

Value	Description
1	No special restrictions.
2	Users will see only their own list items. The WFE MUST NOT allow users without the ManageLists right to update a list item unless the list item was created by that user (for example, **tp_**Author = **@**UserId).
4	Users will not update any list items in this list (1). The WFE MUST NOT allow users without the ManageLists right to add or update list items in this list (1).

tp_Description: The description of this list (1) for display in the WFE.

{tp_Fields}: This field (2) MUST be NULL if the site or list (1) has been flagged to cache all schema data; otherwise, it contains a WSS implementation-specific version string followed by an XML fragment representation of the field definitions of the list (1) as, specified in the **FieldDefinitionDatabaseWithVersion** type defined in section 2.2.8.3.5.

tp_Direction: An enumerated value specifying the direction of text flow for WFE elements presented by this list (1). The following values are valid.

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

AnonymousPermMask: A WSS Rights Mask (section 2.2.2.13) that indicates the rights granted to a user that is anonymous, or has no specific rights, on this list (1).

tp_Flags: A List Flags (section 2.2.2.5) value describing this list (1).

Copyright © 2014 Microsoft Corporation.

tp_ThumbnailSize: The width, in pixels, specified for use when creating thumbnail images of list items within this list (1). This value MUST be NULL for lists that do not have a List Server Template value of 109(Image Library Template). Thumbnails and web images are WFE-generated images of documents that are implementation-specific capabilities.

tp_WebImageWidth: The width, in pixels, specified for use when creating web images of list items within this list (1). This value MUST be NULL for lists that do not have a List Server Template value of 109(Image Library Template).

tp_WebImageHeight: The height, in pixels, specified for use when creating web images of list items within this list (1). This value MUST be NULL for lists that do not have a List Server Template value of 109(Image Library Template).

tp_ImageUrl: The URL of the image used to represent this list (1).

tp_ItemCount: The number of list items that are stored within this list (1).

tp_Author: The identifier of the user who is listed as creating this list (1).

tp_HasInternalFGP: This flag is set to 1 if there have ever been list items for this list (1) that have had a unique ACL applied. Otherwise, this flag is set to 0.

tp_ScopeId: Unique identifier of the security scope for this list (1). This indicates the specific WSS ACL Format ACL (section 2.2.4.6) to use for calculating the permission settings on this list (1).

Acl: The binary serialization of the WSS ACL Format ACL for this list (1). This MUST be used for the permissions for this list (1) if the **tp_ScopeId** value of this list (1) is set to unique scope.

tp_EventSinkAssembly: The name of the .NET assembly that contains the class definition that implements the **event sink** associated with this list (1). This value MUST default to NULL if no event sink has been associated with the list (1).

tp_EventSinkClass: The name of the class definition that implements the event sink associated with this list (1). This value MUST default to NULL if no event sink has been associated with the list (1).

tp_EventSinkData: Additional data persisted on behalf of the event sink implementation, to be passed to the event sink associated with this list (1). This value MUST default to NULL if no event sink has been associated with the list (1).

tp_EmailInsertsFolder: A URL fragment specifying the directory on the configured email inserts server that SHOULD be inspected for new email messages to be processed for this list (1). If the List Flags for this list (1) do not have the value 0x0000000000010000 set, this field (2) MUST be ignored. This value MUST default to NULL if no email inserts server has been associated with the list (1).

tp_EmailInsertsLastSyncTime: This field (2) is a UTC time stamp encoded as a Unicode string in yyyy-mm-dd hh:mi:ss.mmm format specifying the last time the location specified in the **tp_EmailInsertsFolder** column was inspected for new list items. If the List Flags value for this list (1) does not have the value 0x0000000000010000 set, this field (2) MUST be ignored. This value MUST default to NULL if no email inserts server has been associated with the list (1).

tp_EmailAlias: The email alias of the list (1). This alias is used to allow files to be sent directly to this list (1) through an implementation-specific email handling feature. This value MUST default to NULL if no email inserts server has been associated with the list (1).

tp_WebFullUrl: The complete store-relative form URL to the site (2) that contains this list (1).

Copyright © 2014 Microsoft Corporation.

tp_WebId: The Site Identifier (section 2.2.1.11) of the site (2) that contains the list (1).

tp_WebTitle: The title, for display in the WFE, of the site (2) that contains this list (1).

tp_Web: The identifier of the site template for the site (2) that contains this list (1).

tp_WebLanguage: The **language code identifier (LCID)** of the display language of the site (2) that contains this list (1).

tp_WebCollation: The collation order for information in the site (2) that contains this list (1).

tp_SendToLocation: The title and URL for the "Send To Location" configured on this list (1). The "Send To Location" is an implementation-specific feature that allows users to manually save copies of list items and documents to the remote location. This value MUST default to NULL if no "Send To Location" has been associated with the list (1).

{tp_MaxMajorVersionCount}: If the list (1) has versioning enabled, this field (2) contains the number of major versions that will be retained for this document. All versions more than **tp_MaxMajorVersionCount** removed from the current version of the document are automatically removed at version creation time. A value of 0 specifies that versions SHOULD NOT automatically be removed for this list (1).

{tp_MaxMajorwithMinorVersionCount}: If the list (1) has versioning enabled, this field (2) contains the number of major versions that will have their associated minor versions retained for this document. All versions more than **tp_MaxMajorVersionCount** removed from the current version of the document are automatically removed at version creation time. A value of 0 specifies that versions SHOULD NOT automatically be removed for this list (1).

tp_MaxRowOrdinal: This specifies the maximum row ordinal used to store list items for this list (1). This value indicates an implementation-specific calculation for storage of list items within lists.

tp_ListDataDirty: This is set to 1 if the list items in this list (1) require dependency update processing before their next access (for example, updating document link information by parsing each document).

tp_DefaultWorkflowId: The Workflow Identifier (section 2.2.1.16) corresponding to the workflow (2) invoked if the document is in a moderated list (1) and the document is submitted for approval as part of a check in. If the document does not exist, or is not contained in a list (1) with a configured approval Workflow (2), this value MUST be NULL.

tp_ContentTypes: Implementation-specific XML data specifying the content types registered for this list (1).

tp_Subscribed: Set to 1 if an alert for changes to this list (1) has been created in the past, signifying that additional processing needs to be performed.

2.2.5.13 List Web Parts Result Set

The List Web Parts Result Set contains information about the Web parts related to the lists associated with a specified document.

The List Web Parts Result Set is defined using T-SQL syntax, as follows.

```
tp_ListID                   uniqueidentifier,
tp_Type                     tinyint,
tp_Id                       uniqueidentifier,
tp_Flags                    int,
```


Copyright © 2014 Microsoft Corporation.

```
tp_DisplayName                nvarchar(255),
tp_PageUrl                    nvarchar(255),
tp_BaseViewId                 tinyint,
tp_ViewBody                   ntext,
tp_Level                      tinyint,
tp_ContentTypeId              varbinary(512);
```

tp_ListID: The list identifier of the list containing the Web part.

tp_Type: The Page Type (section 2.2.3.14) of the Web part.

tp_Id: The Web Part Identifier (section 2.2.1.15) of the Web part.

tp_Flags: A View Flags value specifying view-related settings for this Web part.

tp_DisplayName: The display name specified for the Web part, if any. This value can be an empty string.

tp_PageUrl: The store-relative form URL to the site (2) that contains this Web part.

tp_BaseViewId: The View Identifier (section 2.2.1.14) for the view where this Web part is used.

tp_ViewBody: An ntext value containing implementation-specific XML used when processing this Web part.

tp_Level: A Publishing Level Type (section 2.2.2.6) value specifying the publishing status of the Web part. This value MUST be NULL if the Web part does not exist.

tp_ContentTypeId: A binary content type identifier (section 2.2.1.1) value specifying the valid content types that this Web part can be used to view. If this Web part is not restricted to a particular content type, this value MUST be 0.

2.2.5.14 NULL Individual URL Security Result Set

The NULL Individual URL Security Result Set indicates that a specified document is not found.

The NULL Individual URL Security Result Set MUST return a single row and is defined using T-SQL syntax, as follows.

```
{ListId}              uniqueidentifier = NULL,
{Acl}                 image = NULL,
{AnonymousPermMask}   bigint = NULL,
{IsAttachment}        bit = 0,
{NeedManageListRight} bit = 0,
{BaseType}            int = NULL,
{ExcludedType}        int = NULL,
{ListFlags}           bigint = NULL,
{Level}               tinyint = NULL,
{DraftOwnerId}        int = NULL,
{DoclibRowId}         int = 0;
```

{ListID}: It MUST be NULL.

{Acl}: It MUST be NULL.

{AnonymousPermMask}: It MUST be NULL.

Copyright © 2014 Microsoft Corporation.

{IsAttachment}: It MUST be 0.

{NeedManagedListRight}: It MUST be 0.

{BaseType}: It MUST be NULL.

{ExcludedType}: It MUST be NULL.

{ListFlags}: It MUST be NULL.

{Level}: It MUST be NULL.

{DraftOwnerId}: It MUST be NULL.

{DoclibRowId}: It MUST be 0.

2.2.5.15 NULL Unique Permissions Result Set

The NULL Unique Permissions Result Set indicates that unique scopes are not found in the specified location. When returned, the result set MUST contain a single row of data. The NULL Unique Permissions Result Set is defined using T-SQL syntax, as follows.

```
{ScopeId}                    binary(1),
{Acl}                        image,
{AnonymousPermMask}          binary(1);
```

{ScopeId}: This MUST be 0x00. This field (2) be cast as a uniqueidentifier with a value of 00000000-0000-0000-0000-000000000000.)

{Acl}: This MUST be NULL.

{AnonymousPermMask}: This MUST be 0x00. This field (2) can be cast as a bigint.

2.2.5.16 Object ID Result Set

The Object ID Result Set contains the GUID of the matching object, if any. The Object ID Result Set is defined using T-SQL syntax, as follows.

```
Id                       uniqueidentifier
```

Id: Contains the GUID of the object specified by the class type, parent object, and name.

2.2.5.17 Principal User Information Result Set

The Principal User Information Result Set returns information about a principal (1). The Principal User Information Result Set is defined using T-SQL syntax, as follows.

```
tp_Id                        int,
tp_SystemID                  varbinary(512),
tp_Title                     nvarchar(255),
tp_Login                     nvarchar(255),
tp_Email                     nvarchar(255),
tp_Notes                     nvarchar(1023),
tp_SiteAdmin                 bit,
```


Copyright © 2014 Microsoft Corporation.

```
tp_DomainGroup                bit;
```

tp_Id: The user identifier of the principal (1). This parameter MUST NOT be NULL.

tp_SystemID: The **SystemID** (section 2.2.1.12) of the principal (1). This parameter MUST NOT be NULL.

tp_Title: Contains the display name of the principal (1). This parameter MUST NOT be NULL.

tp_Login: Contains the login name of the principal (1). This parameter MUST NOT be NULL.

tp_Email: Contains the email address of the principal (1). This parameter MUST NOT be NULL.

tp_Notes: Contains notes associated with the principal (1). This parameter MUST NOT be NULL.

tp_SiteAdmin: Set to 1 if the principal (1) is a **site collection administrator**; otherwise 0. This parameter MUST NOT be NULL.

tp_DomainGroup: Set to 1 if the principal (1) is a domain group; otherwise 0. This parameter MUST NOT be NULL.

2.2.5.18 Server Time Result Set

The Server Time Result Set returns the current time from the BEDS in UTC. The Server Time Result Set is defined using T-SQL syntax, as follows.

```
{CurrentTime}                 datetime;
```

CurrentTime: The current time from the database server, in UTC format.

2.2.5.19 Single Doc Link Information Result Set

The Link Info Result Set returns information about each forward link from, and each backward link to, a specified document or document version.

The Link Info Result Set MUST be ordered by the DocId column and is defined using T-SQL syntax, as follows.

```
DocId                         uniqueidentifier,
LinkDirName                   nvarchar(256),
LinkLeafName                  nvarchar(128),
LinkType                  tinyint,
LinkSecurity                  tinyint,
LinkDynamic                   tinyint,
LinkServerRel                 bit,
LinkStatus                    tinyint,
PointsToDir                   bit,
WebPartId                     uniqueidentifier,
LinkNumber                    int,
WebId                         uniqueidentifier,
Search                        ntext,
FieldId                       uniqueidentifier;
```


Copyright © 2014 Microsoft Corporation.

DocId: The document identifier (section 2.2.1.2) of the specified document.

LinkDirName: Contains the directory name of the link target.

LinkLeafName: Contains the leaf name of the link target.

LinkType: The LinkType (section 2.2.3.10) value of the link. See LinkInfo Types for a list of valid **LinkType** values. This value MUST be NULL for a backward link.

LinkSecurity: The LinkSecurity (section 2.2.3.9) value of the forward link, which specifies whether the scheme of the link is HTTP or HTTPS. This value MUST be NULL for a backward link.

LinkDynamic: The LinkDynamic (section 2.2.3.8) value of the forward link, which specifies whether the link is to a special type of target. This value MUST be NULL for a backward link.

LinkServerRel: A bit flag which specifies whether the link URL is server-relative or not. A value of 1 specifies a server-relative URL. This value MUST be NULL for a backward link.

LinkStatus: The Document Store Type (section 2.2.2.4) value of the document targeted by a forward link. This value MUST be 128 for a backward link. If the forward link target is a document that does not exist, or if this 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 link target is a directory where a welcome page is specified, the link MUST be changed to the URL of the welcome page and PointsToDir MUST be set to 1 so that the link can be distinguished from an explicit link to the welcome page; otherwise, this value MUST be 0.

WebPartId: This value MUST be NULL.

LinkNumber: This value MUST be NULL.

WebId: This value MUST be NULL.

Search: This value MUST be NULL.

FieldId: This value MUST be NULL.

2.2.5.20 Site Audit Mask Result Set

The Site Audit Mask Result Set contains the context-sensitive identifier for a specified object, and the Audit Flags (section 2.2.2.1) set and inherited on that object. The Site Audit Mask Result Set is defined using T-SQL syntax, as follows.

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

{Id}: The identifier for the object. The Audit Item Type (section 2.2.3.2) of the object determines the type of identifier used for this value (for example, a Site Identifier (section 2.2.1.11) or a List Identifier (section 2.2.1.5)). If an object of the specified Audit Item Type at the specified location is found, the value MUST NOT be NULL; otherwise, it MUST be NULL.

Copyright © 2014 Microsoft Corporation.

{AuditFlags}: An Audit Flags value specifying the operations to be audited that are set directly on the specified object. If an object of the specified Audit Item Type at the specified location is not found, the value MUST be NULL.

{InheritAuditFlags}: An Audit Flags value specifying the operations to be audited on the specified object that are inherited from a contained document. If an object of the specified Audit Item Type at the specified location is not found, the value MUST be NULL.

{SiteGlobalAuditMask}: An Audit Flags value specifying the operations to be audited on the specified object that are applied globally within the specified site collection.

2.2.5.21 Site Feature List Result Set

The Site Feature List Result Set contains the list of Feature Identifiers (section 2.2.1.4) of a site (2).

The Site Feature List Result Set is defined using T-SQL syntax, as follows.

```
Feature Iduniqueidentifier;
```

FeatureId: A Feature Identifier for a feature of a site (2).

2.2.5.22 Site Categories Result Set

The Site Categories Result Set contains the categories of a site (2).

The Site Categories Result Set is defined using T-SQL syntax, as follows.

```
Category                    nvarchar(128);
```

Category: The user-friendly name of a category defined for content in the site (2).

2.2.5.23 Site Metadata Result Set

The Site Metadata Result Set contains metadata for a site (2) or sites.

The Site Metadata Result Set is defined using T-SQL syntax, as follows.

```
Id                          uniqueidentifier,
Title                       nvarchar(255),
Description                 ntext,
MetaInfoVersion             int,
WebTemplate                 int,
Language                    int,
Locale                      int,
Collation                   smallint,
TimeZone                    smallint,
Time24                      bit,
CalendarType                smallint,
AdjustHijriDays             smallint,
AltCalendarType             tinyint,
CalendarViewOptions         tinyint,
WorkDays                    smallint,
WorkDayStartHour            smallint,
WorkDayEndHour              smallint,
ProvisionConfig             smallint,
```


Copyright © 2014 Microsoft Corporation.

```
Flags                          int,
Author                         int,
AlternateCSSUrl                nvarchar(260),
CustomizedCss                  nvarchar(260),
CustomJSUrl                    nvarchar(260),
AlternateHeaderUrl             nvarchar(260),
SecurityProvider               uniqueidentifier,
MasterUrl                      nvarchar(260),
CustomMasterUrl                nvarchar(260),
{SiteLogoUrl}                  nvarchar(260),
{SiteLogoDescription}          nvarchar(255),
AnonymousPermMask              bigint,
{SiteFlags}                    int,
{SitePortalURL}                nvarchar(260),
{SitePortalName}               nvarchar(255),
MeetingCount                   smallint,
DefTheme                       nvarchar(64),
CachedNav                      image,
CachedInheritedNav             image,
CachedNavDirty                 int,
CachedDataVersion              int,
NavParentWebId                 uniqueidentifier,
FirstUniqueAncestorWebId       uniqueidentifier,
ScopeId                        uniqueidentifier,
DbNow                          datetime,
Acl                            image,
{RequestAccessEmail}           nvarchar(255),
Ancestry                       image,
ProductVersion                 smallint,
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,
UserId                         int,
{SiteSecurityVersion}          bigint,
tp_Locale                      int,
tp_TimeZone                    smallint,
tp_Time24                      bit,
tp_CalendarType                smallint,
tp_AdjustHijriDays             smallint,
tp_AltCalendarType             tinyint,
tp_CalendarViewOptions         tinyint,
tp_WorkDays                    smallint,
tp_WorkDayStartHour            smallint,
tp_WorkDayEndHour              smallint,
{SiteHashKey}                  binary(16),
{UserInfoListId}               uniqueidentifier,
{RootWebId}                    uniqueidentifier,
{SiteLastContentChange}        datetime,
{SiteLastSecurityChange}       datetime;
```

Id: The Site Identifier (section 2.2.1.8) for the site (2).

Copyright © 2014 Microsoft Corporation.

Title: The title of the site (2) for display in the WFE.

Description: The description of the site (2) for display in the WFE.

MetaInfoVersion: A counter incremented any time a change is made to the Metainfo for this site (2) and used for internal conflict detection.

WebTemplate: The identifier for the template used in the site definition to define the base structure of this site (2).

Language: The LCID associated with the site (2) which is used to determine current UI culture, which determines which language resources to use to display messages on the WFE.

Locale: The LCID associated with the site (2) which is used to determine the current culture for regional language specific data formatting such as currency or date and time settings.

Collation: The Collation Order (section 2.2.3.4) of the site (2) which indicates an additional sorting order that is processed by the BEDS. The collation method is an implementation-specific capability of the WFE and BEDS.

TimeZone: The Time Zone Identifier (section 2.2.3.17) for the time zone to be used when displaying time values for this site (2).

Time24: If set to "1", use a 24-hour time format when displaying time values for this site (2); otherwise, a 12-hour time format can be used.

CalendarType: The Calendar Type (section 2.2.3.3) that to be used when processing date values for this site (2).

AdjustHijriDays: If the Calendar Type value is "6", this specifies the number of days to extend or reduce the current month in Hijri calendars for this site (2).

AltCalendarType: The Calendar Type of an alternate calendar for processing date values for this site. If NULL, only the CalendarType value is used for this site (2).

CalendarViewOptions: A Calendar View Options Type (section 2.2.4.1) that specifies the calendar display options setting for this site (2).

WorkDays: A set of Workdays Flags (section 2.2.2.12) that specify the weekdays defined as the workweek for this site (2).

WorkDayStartHour: The start time of the workday in minutes after midnight for this site (2).

WorkDayEndHour: The end time of the workday in minutes after midnight for this site (2).

ProvisionConfig: An identifier of the site template used to provision this site (2). The following reserved values and implementation-specific site template identifiers are defined.

Value	Description
-1	This site (2) has not had any template provisioned.
0	This site (2) has the implementation-specific default template applied.
1	This site (2) has the Team Collaboration site template applied.
2	This site (2) has the Meeting Workspace site template applied.

Copyright © 2014 Microsoft Corporation.

Value	Description
3	This site (2) has the Central Administration site template applied.
4	This site (2) has the Wiki site template applied.
9	This site (2) has the Blog site template applied.

Flags: A Site Property Flags (section 2.2.2.10) value describing the configuration of this site (2).

Author: The User Identifier (section 2.2.1.13) of the user who is listed as creating this site (2).

AlternateCSSUrl: The URL for a custom CSS sheet file registered on the site for use in pages of the site (2).

CustomizedCss: This contains an implementation-specific list of custom CSS files associated with this site (2).

CustomJSUrl: The URL for a custom JavaScript file registered on the site for use in pages of the site (2).

AlternateHeaderUrl: The URL for a custom header HTML page registered on the site (2) for use in pages of the site when rendered on the WFE.

SecurityProvider: **Component Object Model (COM)** class identifier (CLSID) of the external **security provider** (1) for this site. This is NULL for sites using the native security implementation.

MasterUrl: The URL for the master page registered on the site for use in pages of the site (2) when rendered on the WFE.

CustomMasterUrl: The URL for an alternate master page registered on the site (2) for use in pages of the site rendered on the WFE .

{SiteLogoUrl}: The URL of an image that represents the site (2) for display in the WFE.

{SiteLogoDescription}: The description of the image that represents the site (2) for display in the WFE as an ALT tag on the image.

AnonymousPermMask: A WSS Rights Mask (section 2.2.2.13) that indicates the rights granted to a user that is anonymous, or has no specific rights, to this site (2). The value MUST be NULL for sites that do not exist.

{SiteFlags}: A Site Collection Flags (section 2.2.2.9) value that indicates the settings for the site collection that contains this site (2).

{SitePortalURL}: The URL for a different site registered on the site (2) for use in navigation structures as a parent location.

{SitePortalName}: The display name of a different site registered on the site (2) for use in navigation structures as a parent location.

MeetingCount: If this site is a meeting workspace (that is, its site template identifier in ProvisionConfig is set to 2), this value indicates the number of meetings that are configured.

DefTheme: The name of a theme that is used as part of the display of the site (2).

CachedNav: An implementation-specific serialization of navigation structure information to display in WFE elements associated with this site (2).

Copyright © 2014 Microsoft Corporation.

CachedInheritedNav: This contains an implementation-specific serialization of navigation structure information to display in WFE elements associated with the root navigation elements for this site (2).

CachedNavDirty: If this value is not set to "0", the cached navigation information for this site (2) MUST be regenerated.

CachedDataVersion: A counter incremented on every change to the cached navigation information, used for internal conflict detection.

NavParentWebId: The Site Identifier of the site (2) to be treated as the parent of this site when displaying navigation elements in the WFE.

FirstUniqueAncestorWebId: The Site Identifier of the closest site (2) in this site's ancestor chain that does not inherit security settings from its parent site.

ScopeId: The Scope Identifier (section 2.2.1.8) of the scope containing the site (2).

DbNow: The current time in UTC format on the BEDS.

Acl: The binary serialization of the WSS ACL Format (section 2.2.4.6) ACL for this site (2). This is either explicitly defined or inherited from the parent object of the site.

{RequestAccessEmail}: The email address for a user who has the authority to grant access to the site (2), if any. This can be used by the WFE to generate an email to request access for a user who does not have access to the site.

Ancestry: An implementation-specific serialization of the navigation structure information for the site (2).

ProductVersion: The version of Windows SharePoint Services used to create the site (2).

tp_Id: The User Identifier of the current user.

tp_SiteAdmin: Set to "1" if the current user is an administrator of the site (2).

tp_IsActive: Set to "1" if the current user is an active user in the site collection containing this site (2).

tp_Login: The login information of the current user.

tp_Email: The email address of the current user.

tp_Title: The display name of the current user.

tp_Notes: Notes about the current user.

tp_ExternalTokenLastUpdated: A time stamp in UTC format specifying the time when the External Group Token (section 2.2.4.2) for the current user was last updated.

tp_Token: A WSS User Token (section 2.2.4.9) value specifying the site group membership of the current user.

UserId: The site membership identifier of the current user. This is NULL if the current user has not been added to the set of members of this site (2).

{SiteSecurityVersion}: The current security information version of the site collection.

Copyright © 2014 Microsoft Corporation.

tp_Locale: An LCID referring to the locale value to be used when displaying messages in the WFE for the current user. If this value is NULL, the site setting for Locale is used instead.

tp_TimeZone: The Time Zone Identifier for the time zone to be used when displaying time values for the current user. If this value is NULL, the site setting for TimeZone is used instead.

tp_Time24: A bit flag which specifies whether to use a 24-hour time format when displaying time values to the current user. If this parameter is set to "1", the 24-hour time format is be used; otherwise, the 12-hour time format is be used. If this value is NULL, the site setting for Time24 MUST be used instead.

tp_CalendarType: The Calendar Type to be used when processing date values for the current user. If this value is NULL, the site setting for CalendarType MUST be used instead.

tp_AdjustHijriDays: If the **tp_CalendarType** value is "6", this specifies the number of days to extend or reduce the current month in Hijri calendars for the current user. If this value is NULL, the site setting for AdjustHijriDays MUST be used instead.

tp_AltCalendarType: The Calendar Type of an alternate calendar to be used when displaying calendars in views for the current user. If this value is NULL, the site setting for **AltCalendarType** MUST be used instead.

tp_CalendarViewOptions: A Calendar View Options Type specifying the calendar options for the current user. If this value is NULL, the site setting for CalendarViewOptions MUST be used instead.

tp_WorkDays: A set of Workdays Flags specifying the weekdays defined as the workweek for the current user. If this value is NULL, the site setting for WorkDays MUST be used instead.

tp_WorkDayStartHour: The start time of the work day in minutes after midnight for the current user. If this value is NULL, the site setting for WorkDayStartHour MUST be used instead.

tp_WorkDayEndHour: The end time of the workday in minutes after midnight for the current user. If this value is NULL, the site setting for **WorkDayEndHour** MUST be used instead.

{SiteHashKey}: Contains binary information used when generating digital signatures of information associated with this site collection.

{UserInfoListId}: The List Identifier (section 2.2.1.5) of the user information list for the site collection containing this site (2).

{RootWebId}: The Site Identifier for the root site (2) of the site collection containing this site.

{SiteLastContentChange}: A time stamp in UTC format describing the time content was last changed within this site collection.

{SiteLastSecurityChange}: A time stamp in UTC format describing the time security settings were last changed within this site collection.

2.2.5.24 Site MetaInfo Result Set

The Site MetaInfo Result Set returns metainfo about a site (2).

The Site MetaInfo Result Set is defined using T-SQL syntax, as follows.

```
MetaInfo                    image;
```


Copyright © 2014 Microsoft Corporation.

MetaInfo: Site Metainfo in the form of a metadict. This value MAY be NULL. The metadict format is specified in [MS-FPSE] section 2.2.2.2.11.

2.2.5.25 Unique Permissions Result Set

The Unique Permissions Result Set returns the ACL information for all the unique scopes of folders and list items contained in a specified location.

The Unique Permissions Result Set is defined using T-SQL syntax, as follows.

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

ScopeId: The Scope Identifier (section 2.2.1.8) of the scope.

Acl: The binary serialization of the WSS ACL Format ACL of the scope.

AnonymousPermMask: A WSS Rights Mask (section 2.2.2.13) that indicates the rights granted to an anonymous user, or a user who has no specific rights on this scope.

2.2.5.26 URL Result Set

The URL Result Set returns a list of URLs for subsites of a site (2). The URL Result Set is defined using T-SQL syntax as follows.

```
FullUrl                     nvarchar (256)
```

FullUrl: The store-relative form URL of the Subsite.

2.2.6 SQL Structures

2.2.6.1 Configuration Object

A Configuration Object encapsulates as a group of application settings a set of metadata used to identify and manipulate those settings. Configuration Objects have the following fields.

2.2.6.1.1 Class

Applications use a Configuration Object Class to help distinguish the schemas of the data stored in different objects. Class is a complex data type with the following fields, defined in T-SQL syntax.

```
ClassId                     uniqueidentifier NOT NULL,
BaseClassId                 uniqueidentifier NOT NULL,
FullName                    nvarchar(256)    NOT NULL
```

ClassId: This MUST be a GUID that is different from all other ClassIds registered in the Configuration Database.

BaseClassId: A class is said to derive from the class referenced by BaseClassId. This allows a class hierarchy to be built. Many ClassIds can share a single BaseClassId. The BaseClassId of a Configuration Object Class MUST have the same value as a ClassId in the Configuration Database. If a Configuration Object Class has no base class, its BaseClassId MUST be identical to its ClassId.

Copyright © 2014 Microsoft Corporation.

FullName: An identifier that can be used by an application to associate a ClassId with a human-readable or machine-readable string.

The following classes are used during the execution of this protocol.

Class	ClassId
Alternate URL Collection	9920F486-2FF4-4d10-9532-E01979826585
Content Database	3D4F5451-1735-48bb-B920-76C1EC240B1D
Database Service Instance	3112E92F-B97D-481e-8CEB-03FDE15ED1A7
Farm	674DA553-EA77-44A3-B9F8-3F70D786DE6A
Server	E77AAF47-3CAC-4001-BC6B-5BCCB6486318
Web Application	113FB569-7520-4651-8FC4-E9F4F5887618
Web Service	45AD2BF2-4E3E-46A1-B477-126944C0ACEF

2.2.6.1.2 Id

A Configuration Object's (section 2.2.6.1) Id is used to identify a single instance of a Configuration Object. As such, it MUST be different from the IDs of all other Configuration Objects in the Configuration Database. Id is defined in T-SQL format as follows.

```
Id                              uniqueidentifier NOT NULL
```

2.2.6.1.3 Name

Configuration Objects (section 2.2.6.1) can also be identified by name. Unlike Id, a Configuration Object's name is only used to distinguish instances of Configuration Objects with the same parent and class. The combination of name, parent, and class MUST be unique among the set of Configuration Objects in a single Configuration Database. Name is defined in T-SQL format as follows.

```
Name                        nvarchar(128) NOT NULL
```

2.2.6.1.4 Parent

Configuration Objects (section 2.2.6.1) are organized in an ancestry hierarchy. Each Configuration Object MUST have one and only one parent, which MUST be a Configuration Object registered in the same Configuration Database. The set of Configuration Objects with a given parent are known as the parent's children. A Configuration Object's descendants are the set of Configuration Objects including the parent's children, the children's children, and so on.

The Configuration Object at the root of an ancestry hierarchy MUST be its own parent.

The Configuration Database MUST delete a Configuration Object when its parent is deleted.

A Configuration Object defines its parent through its own ParentId property, which is the Id of its parent. ParentId is defined in T-SQL format as follows.

Copyright © 2014 Microsoft Corporation.

```
ParentId                        uniqueidentifier NOT NULL
```

2.2.6.1.5 Status

The Status property of a Configuration Object (section 2.2.6.1) MUST conform to the data type defined in the Configuration Object Status section (section 2.2.2.2).

```
Status          int NOT NULL
```

2.2.6.1.6 Version

Version is used to identify when a Configuration Object (section 2.2.6.1) has been created, modified, or deleted. Version is defined in T-SQL format as follows.

```
Version                         rowversion NOT NULL
```

2.2.6.1.7 Properties

Any configuration settings needed by an application that are not stored in one of the other Configuration Object (section 2.2.6.1) field (2) can be stored in the properties field, which is defined using T-SQL format as follows.

```
Properties                      ntext
```

Successful execution requires parsing the properties of the following classes. Each of these classes stores its properties in an XML format. Because only a portion of each class' properties field (2) is needed for successful execution, the values needed are identified using queries from the XML Path Language specified in [XPATH]. The schemata of the contents of other Configuration Objects' Property Bags will differ based on the needs of each application and are not specified in this document. Unless otherwise noted in the description column, each XPath query resolves to one and only one element value.

2.2.6.1.7.1 Alternate URL Collection

An Alternate URL Collection Configuration Object (section 2.2.6.1) stores a list of absolute URLs. The parent of an Alternate URL Collection MUST be a Farm Configuration Object.

Property	XPath Query	Description
Alternate URL	/object/fld[attribute::name='m_Urls']/fld/object/sFld[attribute::name='m_RequestUri']	When executed against the properties of an Alternate URL Collection Configuration Object, this query MUST return one or more values containing absolute URLs. The URLs MUST contain only the portion of the incoming URL

Copyright © 2014 Microsoft Corporation.

Property	XPath Query	Description
		beginning with the scheme component and ending with the Authority Component (for example, "http://example.com:80"). The URLs MUST be unique within a Configuration Database.

2.2.6.1.7.2 Content Database

The Content Database Configuration Object stores information needed to establish a connection to a content database.

Property	XPath Query	Description
Username	/object/fld[attribute::name='m_Username']	If the value returned by this XPath query is not NULL or empty, the client uses SQL authentication to connect to the content database. The client MUST pass the value returned by the XPath query as the SQL authentication username.
Password	/object/fld[attribute::name='m_Password']	If the value returned by this XPath query is not NULL or empty, the client MUST include this value in the user name portion of the content database connection string.

2.2.6.1.7.3 Web Application

The Web Application Configuration Object stores a list of content databases used by the **web application** (1) as well as information needed to parse URLs. The parent of a Web Application Configuration Object MUST be a Web Service Configuration Object.

Property	XPath Query	Description
Web Application Alternate URL Collection	/object/fld[attribute::name='m_AlternateUrlCollection’]	This value is a string representation of a Configuration Object ID. It is used to associate a web application (1) with an Alternate URL Collection. Each Alternate URL Collection MUST be referenced by zero or one

Copyright © 2014 Microsoft Corporation.

Property	XPath Query	Description
		web applications (1).
Prefix Name	/object/fld[attribute::name='m_Prefixes']/object/fld/fld/object/sFld[attribute::name='m_Name']	The value returned by this query MUST be a valid URL Path segment.
Prefix Type	/object/fld[attribute::name='m_Prefixes']/object/fld/fld/object/sFld[attribute::name='m_Type']	The value returned by this query MUST be the literal "ExplicitInclusion" or the literal "WildcardInclusion".

2.2.6.2 Dependencies

In addition to the ancestry hierarchy defined in Parent section (section 2.2.6.1.4), Configuration Objects (section 2.2.6.1) can also define Dependencies. Dependencies can be used by an application to discover the list of Configuration Objects that depend on a given Configuration Object.

A Dependency is a complex type with the following fields, defined in T-SQL format as follows.

```
ObjectId                    uniqueidentifier NOT NULL
DependantId                 uniqueidentifier NOT NULL
```

ObjectId: The ID of the Configuration Object that depends on another.

DependantId: The ID of the Configuration Object upon which the Configuration Object identified by **ObjectId** depends.

2.2.7 Tables and Views

2.2.7.1 AllDocs Table

The **AllDocs** Table stores data for all documents in the SharePoint Store. The **AllDocs** Table is defined using T-SQL syntax as follows.

```
TABLE AllDocs(
Id                          uniqueidentifier  NOT NULL,
SiteId                      uniqueidentifier  NOT NULL,
DirName                     nvarchar(256)     NOT NULL,
LeafName                    nvarchar(128)     NOT NULL,
WebId                       uniqueidentifier  NOT NULL,
ListId                      uniqueidentifier  NULL,
DoclibRowId                 int               NULL,
Type                        tinyint           NOT NULL,
Size                        int               NULL,
MetaInfoSize                int               NULL,
Version                     int               NULL,
```


Copyright © 2014 Microsoft Corporation.

```
UIVersion                       int               NOT NULL DEFAULT 512,
Dirty                           bit               NULL,
ListDataDirty                   bit               NOT NULL DEFAULT 0,
CacheParseId                    uniqueidentifier  NULL,
DocFlags                        int               NULL,
ThicketFlag                     bit               NULL DEFAULT 0,
CharSet                         int               NULL,
ProgId                          nvarchar(255)     NULL,
TimeCreated                     datetime          NOT NULL,
TimeLastModified                datetime          NOT NULL,
NextToLastTimeModified          datetime          NULL,
MetaInfoTimeLastModified        datetime          NULL,
TimeLastWritten                 datetime          NULL,
DeleteTransactionId             varbinary(16)     NOT NULL DEFAULT 0x,
SetupPathVersion                tinyint           NOT NULL DEFAULT 3,
SetupPath                       nvarchar(255)     NULL,
SetupPathUser                   nvarchar(255)     NULL,
CheckoutUserId                  int               NULL,
CheckoutDate                    datetime          NULL,
CheckoutExpires                 datetime          NULL,
VersionCreatedSinceSTCheckout   bit               NOT NULL DEFAULT 0,
LTCheckoutUserId                AS                (case when (DocFlags&(32))=(32)
                                                  then CheckoutUserId  end),
VirusVendorID                   int               NULL,
VirusStatus                     int               NULL,
VirusInfo                       nvarchar(255)     NULL,
MetaInfo                        image             NULL,
UnVersionedMetaInfo             image             NULL,
UnVersionedMetaInfoSize         int               NULL,
UnVersionedMetaInfoVersion      int               NULL,
WelcomePageUrl                  nvarchar(260)     NULL,
WelcomePageParameters           ntext             NULL,
IsCurrentVersion                bit               NOT NULL DEFAULT 1,
Level                           tinyint           NOT NULL DEFAULT 1,
CheckinComment                  nvarchar(1023)    NULL,
AuditFlags                      int               NULL,
InheritAuditFlags               int               NULL,
DraftOwnerId                    int               NULL,
UIVersionString                 AS                ((CONVERT(nvarchar,UIVersion
                                                          /(512),0)+'.')
                                                  +CONVERT(nvarchar,
                                                         UIVersion%(512),0)),
ParentId                        uniqueidentifier  NOT NULL,
HasStream                       AS                (case when Type=(0) AND
                                                  (DocFlags&(256))=(256) AND
                                                  SetupPath IS NULL OR SetupPath
                                                  IS NOT NULL AND
                                                  (DocFlags&(64))=(64) then (1)
                                                          else (0) end),
ScopeId                         uniqueidentifier  NOT NULL,
BuildDependencySet              image             NULL,
ParentVersion                   int               NULL,
ParentVersionString             AS                ((CONVERT(nvarchar,ParentVersion/
                                                          (512),0)+'.')
                                                  +CONVERT(nvarchar,ParentVersion%
                                                          (512),0)),
TransformerId                   uniqueidentifier  NULL,
ParentLeafName                  nvarchar(128)     NULL,
IsCheckoutToLocal               AS               (case when (DocFlags&(512))=(512)
```


Copyright © 2014 Microsoft Corporation.

```
                                                then (1) else (0) end),
  CtoOffset                     smallint          NULL,
  Extension                     AS               (case when charindex(N'.',
                                                  LeafName collate
                                                  Latin1_General_BIN)>(0)
                                                  then right(LeafName,charindex(N'.',
                                                  reverse(LeafName)
                                                  collate Latin1_General_BIN)-(1))
                                                  else N'' end),
  ExtensionForFile              AS               (case when Type=(0) AND charindex(N'.',
                                                  LeafName collate Latin1_General_BIN)>(0)
                                                  then right(LeafName,charindex(N'.',
                                                  reverse(LeafName) collate
                                                  Latin1_General_BIN)-(1))
                                                  else N'' end),
  ItemChildCount                int               NOT NULL DEFAULT 0,
  FolderChildCount              int               NOT NULL DEFAULT 0,
  );
```

Id: The document identifier (section 2.2.1.2) of the document.

SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the document.

DirName: The directory name of the document location.

LeafName: The leaf name of the document location.

WebId: The Site Identifier (section 2.2.1.11) of the site containing the document.

ListId: The List Identifier (section 2.2.1.5) of the list (1) containing the document, if any, or NULL if the document is not contained in a list (1).

DoclibRowId: The row identifier for the document within the containing document library or list (1), if applicable.

Type: An integer identifier specifying the document's Document Store Type (section 2.2.2.4).

Size: The size of the document stream in bytes. This parameter can be set to NULL or to 0 for items such as sites, folders, document libraries, and lists.

MetaInfoSize: The size, in bytes, of the document metadata info.

Version: A counter incremented any time a change is made to this document and used for internal conflict detection. This is an internal document version separate from the user-visible versioning system reflected in **UIVersion** and **UIVersionString**.

UIVersion: A UI version number associated with the document. UIVersion defaults to 512, which corresponds to a displayed version of 1.0.

Dirty: Set to 1 if the document has dependencies that require further processing.

ListDataDirty: Set to 1 if the document requires subsequent fix-up of link information.

CacheParseId**:** A SharePoint implementation-specific identifier for an internal dependency update **process**. Used to manage bulk updates to documents when processed for dependency fix-up.

Copyright © 2014 Microsoft Corporation.

DocFlags: A Doc Flags (section 2.2.2.3) value specifying information about the document. See the Doc Flags section for details.

ThicketFlag: If set to 1, indicates that the document is an auxiliary thicket file, one of a set of files supporting a thicket.

CharSet: An optional character set associated with the document. Any **Windows code page** identifier is valid for **CharSet**. This value can be NULL, indicating that no windows code page is specified for the document.

ProgId: An OPTIONAL preferred application to open this document. The **ProgId** is used to distinguish between different applications that save files with a given file extension (that is, different editors for HTML or XML files).

TimeCreated: A time stamp in UTC format specifying when this document was created.

TimeLastModified: A time stamp in UTC format. The value specifies when the document was last saved. This corresponds to the actual time when the document was last modified.

NextToLastTimeModified: The value of **TimeLastModified** from the previous time that the document was saved.

MetaInfoTimeLastModified: A time stamp in UTC format specifying when the Metadata information for the document was last changed.

TimeLastWritten: A time stamp in UTC format specifying when any changes were made to the document stream. This does not reflect changes made strictly to the **MetaInfo** or other item properties.

DeleteTransactionId: The transaction identifier for the implementation-specific deleted items recycle bin. A value of 0x specifies that this document is not marked as deleted. In the Docs View (section 2.2.7.4), **DeleteTransactionId** MUST equal 0x, because the Docs View only displays items that are not marked as deleted.

SetupPathVersion: For a ghosted document, this parameter governs the **SetupPath** fragment relative to the setup path location. The following values are valid.

Value	Description
NULL	This is NOT a ghosted document.
2	The SetupPath is relative to the install location of Windows SharePoint Services 2.0 on the WFE (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\60)
3	The SetupPath is relative to the install location of Windows SharePoint Services 3.0 on the WFE (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\12).

SetupPath: For a document that is now or once was ghosted, this contains the setup path fragment relative to the base setup path specified by the **SetupPathVersion** value, where the content stream of this document can be found. This value MUST be NULL if the document was never ghosted.

SetupPathUser: If this document is now or once was ghosted, this contains the login name of the user who created the ghosted document. This value is undefined for documents that were never ghosted.

Copyright © 2014 Microsoft Corporation.

CheckoutUserId: If the document is checked out, this parameter contains the User Identifier (section 2.2.1.13) of the user who has the document checked out. Otherwise, this parameter is NULL.

CheckoutDate: A time stamp in UTC format indicating when this document was checked out.

CheckoutExpires: A time stamp in UTC format indicating when the short-term lock for this document will expire.

VersionCreatedSinceSTCheckout: If this parameter is 1, the document version has been incremented since the document last had a short-term lock established. This is used to prevent more than one new version of the document from being created while a short-term lock is established.

LTCheckoutUserId: If the document is currently checked out, this parameter is a calculated column containing the value of **CheckoutUserId**. Otherwise, this parameter is NULL.

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 check status of this document. This value MUST be NULL if the document has not been processed by a virus scanner. See Virus Status (section 2.2.3.18) in the Flags section 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. This value MUST be NULL if the document has not been processed by a virus scanner.

MetaInfo: A metadict for the document. The metadict format is specified in the [MS-FPSE] metadict section.

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

UnVersionedMetaInfoSize: The size in bytes of **UnVersionedMetaInfo**.

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

WelcomePageUrl: If the document is a folder, this optionally specifies a page to redirect to when the folder is requested with an HTTP GET operation. The URL is relative to the URL of the folder itself and MUST be subsumed by that folder. Attempts to break out of the folder, such as "../../default.aspx", are not valid.

WelcomePageParameters: This parameter MUST contain any URL parameters configured for the welcome page. This value contains a query string starting with "?" or a hash parameter starting with "#".

IsCurrentVersion: If this value is set to 1, this row is for a current version of the document (out of all the rows corresponding to the given **DirName**, **LeafName**, each with a different Publishing Level Type (section 2.2.2.6) value). Otherwise, this row is for a historical version of the document.

Level: A Publishing Level Type value specifying the publishing level of this version of the document as checked out, draft, or published.

CheckinComment: An OPTIONAL comment string associated with the document when it was last checked in or published.

Copyright © 2014 Microsoft Corporation.

AuditFlags: An Audit Flags (section 2.2.2.1) value determining the operations to be tracked on this document.

InheritAuditFlags: An Audit Flags value determining the operations to be tracked on this document's children.

DraftOwnerId: If the current publishing level of the document is a draft, this is the User Identifier of the user who has that draft version checked out. Otherwise, this MUST be set to NULL.

UIVersionString: A calculated column presenting the value of the **UIVersion** column as a displayed version string, in the following format**:** major version value, followed by '.', followed by the minor version value. For example, **UIVersion** values 512, 513, 514, and 1024 correspond to **UIVersionString** values of 1.0, 1.1, 1.2, and 2.0, respectively.

ParentId: The document identifier (2) of the document's parent container. For example, the document identifier (2) of the folder or document library containing this subfolder or document. This value MUST NOT be NULL, because every item but one has a parent container. A special empty document identifier (2), '00000000-0000-0000-0000-000000000000', marks the parent container of the topmost item (the root site) in the site collection.

HasStream: A calculated bit indicating whether the document has an associated document stream. This MUST be set to "1" if**:**

- The **SetupPath** is NULL and the document's Document Store Type is 0 (file) and it is allowed to contain a stream.
- The **SetupPath** is not NULL and the document content is **UnGhosted**.

If neither condition applies, the value MUST be "0".

ScopeId: The Scope Identifier (section 2.2.1.8) of the scope of the document.

BuildDependencySet: A binary image holding implementation-specific information about dependent documents. NULL indicates that there is no information about dependencies. A **BuildDependencySet** of size 0 indicates a document with no dependencies.

ParentVersion: If the document is a transformed version of another document, this is the UIVersion value from the original document. This value MUST be NULL if the document is not the product of a document transformation.

ParentVersionString: A calculated column presenting the value of the **ParentVersion** column as a displayed version string. See **UIVersionString** for displayed version format and examples.

TransformerId: If the document is a transformed version of another document, this MUST be the GUID of the agent that performed the transformation. Otherwise, **TransformerId** MUST be NULL.

ParentLeafName: If the document is a transformed version of another document, this is the leaf name of the original document. The original document MUST be in the same folder as the transformed document. This value MUST be NULL if the document is not the product of a document transformation.

IsCheckoutToLocal: A calculated bit indicating whether the document is checked out to a client's local disk. This parameter MUST be set to "1" if the **DocFlags** bit array includes 512 (long-term check-out to local); otherwise, it MUST be set to "0".

Copyright © 2014 Microsoft Corporation.

CtoOffset: A content type order offset. This is a SharePoint implementation-specific value used by list and document library views to allow custom ordering of the menu items under the 'New' button. This value can be NULL.

Extension: A calculated field (2) holding the file extension part, if applicable, from the LeafName value. If the **LeafName** value contains a "." character, Extension MUST hold the string following the last "."; otherwise, Extension MUST be set to the empty string. For example, if the full leaf name is 'document.docx.bac', Extension MUST be "bac".

ExtensionForFile: A calculated field (2) holding the file extension part, if applicable, from the **LeafName** value. If the document's Document Store Type is set to "0", indicating that it is a file type, **ExtensionForFile** MUST be identical to Extension; otherwise, **ExtensionForFile** value MUST be an empty string.

ItemChildCount: The number of nonfolder children of this document.

FolderChildCount: The number of folder children of this document.

2.2.7.2 AllDocStreams Table

The **AllDocStreams** Table stores the document stream and related data for documents with content streams stored in the content database on the BEDS. The **AllDocStreams** Table is defined using T-SQL syntax, as follows.

```
TABLE AllDocStreams(
      Id                          uniqueidentifier   NOT NULL,
      SiteId                      uniqueidentifier   NOT NULL,
      DeleteTransactionId         varbinary(16)      NOT NULL   DEFAULT (0x),
      ParentId                    uniqueidentifier   NOT NULL   DEFAULT ('00000000-0000-
0000-0000-000000000000'),
      Size                        int                NULL,
      Level                       tinyint            NOT NULL   DEFAULT (1),
      Content                     image              NULL
);
```

Id: The document identifier (section 2.2.1.2) of the document.

SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the document.

DeleteTransactionId: The transaction identifier for the implementation-specific deleted items **Recycle Bin**. A value of 0x specifies that this document is not marked as deleted.

ParentId: The document identifier (2) of the document's parent container. This value MUST NOT be NULL, because every item in a site collection but one has a parent container. A special empty document identifier (2), '00000000-0000-0000-0000-000000000000', specifies the parent container of the topmost item (the root site) in the site collection.

Size: The size of the document stream, in bytes. For a ghosted document, this can be NULL.

Level: A Publishing Level Type (section 2.2.2.6) value specifying the publishing level of this version of the document as checked out, draft, or published.

Content: The document stream with the content of the document. For a ghosted document, this can be NULL.

Copyright © 2014 Microsoft Corporation.

2.2.7.3 AllUserData Table

The **AllUserData** Table stores data for all list (1) and document library list items. The table provides a fixed number of generic columns in various data types, affording storage for application-defined variable schemas. A list item can be represented by more than one row in this table, if its list's schema requires more entries of a particular data type than are available in a single row. Application-defined metadata for documents in document libraries also resides in **AllUserData**, and it is accessed via joins with the Docs View (section 2.2.7.4).

The table is defined using T-SQL syntax, as follows.

```
TABLE AllUserData (
tp_Id                       int                NOT NULL,
tp_ListId                   uniqueidentifier   NOT NULL,
tp_SiteId                   uniqueidentifier   NOT NULL,
tp_RowOrdinal               int                NOT NULL   DEFAULT ((0)),
tp_Version                  int                NOT NULL,
tp_Author                   int                NULL,
tp_Editor                   int                NULL,
tp_Modified                 datetime           NULL,
tp_Created                  datetime           NULL,
tp_Ordering                 varchar(512)       NULL,
tp_ThreadIndex              varbinary(512)     NULL,
tp_HasAttachment            bit                NOT NULL   DEFAULT ((0)),
tp_ModerationStatus         int                NOT NULL   DEFAULT ((0)),
tp_IsCurrent                bit                NOT NULL   DEFAULT ((1)),
tp_ItemOrder                float              NULL,
tp_InstanceID               int                NULL,
tp_GUID                     uniqueidentifier   NOT NULL   DEFAULT (newid()),
tp_CopySource               nvarchar(260)      NULL       DEFAULT (NULL),
tp_HasCopyDestinations      bit                NULL       DEFAULT ((0)),
tp_AuditFlags               int                NULL,
tp_InheritAuditFlags        int                NULL,
tp_Size                     int                NOT NULL   DEFAULT ((0)),
tp_WorkflowVersion          int                NULL,
tp_WorkflowInstanceID       uniqueidentifier   NULL,
tp_DirName                  nvarchar(256)      NOT NULL,
tp_LeafName                 nvarchar(128)      NOT NULL,
tp_DeleteTransactionId      varbinary(16)      NOT NULL   DEFAULT (0x),
tp_ContentType              nvarchar(255)      NULL,
tp_ContentTypeId            varbinary(512)     NULL,
nvarchar1                   nvarchar(255)      NULL,
nvarchar2                   nvarchar(255)      NULL,
nvarchar3                   nvarchar(255)      NULL,
nvarchar4                   nvarchar(255)      NULL,
nvarchar5                   nvarchar(255)      NULL,
nvarchar6                   nvarchar(255)      NULL,
nvarchar7                   nvarchar(255)      NULL,
nvarchar8                   nvarchar(255)      NULL,
ntext1                      ntext              NULL,
ntext2                      ntext              NULL,
ntext3                      ntext              NULL,
ntext4                      ntext              NULL,
sql_variant1                sql_variant        NULL,
nvarchar9                   nvarchar(255)      NULL,
nvarchar10                  nvarchar(255)      NULL,
nvarchar11                  nvarchar(255)      NULL,
nvarchar12                  nvarchar(255)      NULL,
```


Copyright © 2014 Microsoft Corporation.

```
  nvarchar13                    nvarchar(255)      NULL,
  nvarchar14                    nvarchar(255)      NULL,
  nvarchar15                    nvarchar(255)      NULL,
  nvarchar16                    nvarchar(255)      NULL,
  ntext5                        ntext              NULL,
  ntext6                        ntext              NULL,
  ntext7                        ntext              NULL,
  ntext8                        ntext              NULL,
  sql_variant2                  sql_variant        NULL,
  nvarchar17                    nvarchar(255)      NULL,
  nvarchar18                    nvarchar(255)      NULL,
  nvarchar19                    nvarchar(255)      NULL,
  nvarchar20                    nvarchar(255)      NULL,
  nvarchar21                    nvarchar(255)      NULL,
  nvarchar22                    nvarchar(255)      NULL,
  nvarchar23                    nvarchar(255)      NULL,
  nvarchar24                    nvarchar(255)      NULL,
  ntext9                        ntext              NULL,
  ntext10                       ntext              NULL,
  ntext11                       ntext              NULL,
  ntext12                       ntext              NULL,
  sql_variant3                  sql_variant        NULL,
  nvarchar25                    nvarchar(255)      NULL,
  nvarchar26                    nvarchar(255)      NULL,
  nvarchar27                    nvarchar(255)      NULL,
  nvarchar28                    nvarchar(255)      NULL,
  nvarchar29                    nvarchar(255)      NULL,
  nvarchar30                    nvarchar(255)      NULL,
  nvarchar31                    nvarchar(255)      NULL,
  nvarchar32                    nvarchar(255)      NULL,
  ntext13                       ntext              NULL,
  ntext14                       ntext              NULL,
  ntext15                       ntext              NULL,
  ntext16                       ntext              NULL,
  sql_variant4                  sql_variant        NULL,
  nvarchar33                    nvarchar(255)      NULL,
  nvarchar34                    nvarchar(255)      NULL,
  nvarchar35                    nvarchar(255)      NULL,
  nvarchar36                    nvarchar(255)      NULL,
  nvarchar37                    nvarchar(255)      NULL,
  nvarchar38                    nvarchar(255)      NULL,
  nvarchar39                    nvarchar(255)      NULL,
  nvarchar40                    nvarchar(255)      NULL,
  ntext17                       ntext              NULL,
  ntext18                       ntext              NULL,
  ntext19                       ntext              NULL,
  ntext20                       ntext              NULL,
  sql_variant5                  sql_variant        NULL,
  nvarchar41                    nvarchar(255)      NULL,
  nvarchar42                    nvarchar(255)      NULL,
  nvarchar43                    nvarchar(255)      NULL,
  nvarchar44                    nvarchar(255)      NULL,
  nvarchar45                    nvarchar(255)      NULL,
  nvarchar46                    nvarchar(255)      NULL,
  nvarchar47                    nvarchar(255)      NULL,
  nvarchar48                    nvarchar(255)      NULL,
  ntext21                       ntext              NULL,
  ntext22                       ntext              NULL,
  ntext23                       ntext              NULL,
```


Copyright © 2014 Microsoft Corporation.

```
ntext24                         ntext               NULL,
sql_variant6                    sql_variant         NULL,
nvarchar49                      nvarchar(255)       NULL,
nvarchar50                      nvarchar(255)       NULL,
nvarchar51                      nvarchar(255)       NULL,
nvarchar52                      nvarchar(255)       NULL,
nvarchar53                      nvarchar(255)       NULL,
nvarchar54                      nvarchar(255)       NULL,
nvarchar55                      nvarchar(255)       NULL,
nvarchar56                      nvarchar(255)       NULL,
ntext25                         ntext               NULL,
ntext26                         ntext               NULL,
ntext27                         ntext               NULL,
ntext28                         ntext               NULL,
sql_variant7                    sql_variant         NULL,
nvarchar57                      nvarchar(255)       NULL,
nvarchar58                      nvarchar(255)       NULL,
nvarchar59                      nvarchar(255)       NULL,
nvarchar60                      nvarchar(255)       NULL,
nvarchar61                      nvarchar(255)       NULL,
nvarchar62                      nvarchar(255)       NULL,
nvarchar63                      nvarchar(255)       NULL,
nvarchar64                      nvarchar(255)       NULL,
ntext29                         ntext               NULL,
ntext30                         ntext               NULL,
ntext31                         ntext               NULL,
ntext32                         ntext               NULL,
sql_variant8                    sql_variant         NULL,
int1                            int                 NULL,
int2                            int                 NULL,
int3                            int                 NULL,
int4                            int                 NULL,
int5                            int                 NULL,
int6                            int                 NULL,
int7                            int                 NULL,
int8                            int                 NULL,
int9                            int                 NULL,
int10                           int                 NULL,
int11                           int                 NULL,
int12                           int                 NULL,
int13                           int                 NULL,
int14                           int                 NULL,
int15                           int                 NULL,
int16                           int                 NULL,
float1                          float               NULL,
float2                          float               NULL,
float3                          float               NULL,
float4                          float               NULL,
float5                          float               NULL,
float6                          float               NULL,
float7                          float               NULL,
float8                          float               NULL,
float9                          float               NULL,
float10                         float               NULL,
float11                         float               NULL,
float12                         float               NULL,
datetime1                       datetime            NULL,
datetime2                       datetime            NULL,
datetime3                       datetime            NULL,
```


Copyright © 2014 Microsoft Corporation.

```
  datetime4                     datetime          NULL,
  datetime5                     datetime          NULL,
  datetime6                     datetime          NULL,
  datetime7                     datetime          NULL,
  datetime8                     datetime          NULL,
  bit1                          bit               NULL,
  bit2                          bit               NULL,
  bit3                          bit               NULL,
  bit4                          bit               NULL,
  bit5                          bit               NULL,
  bit6                          bit               NULL,
  bit7                          bit               NULL,
  bit8                          bit               NULL,
  bit9                          bit               NULL,
  bit10                         bit               NULL,
  bit11                         bit               NULL,
  bit12                         bit               NULL,
  bit13                         bit               NULL,
  bit14                         bit               NULL,
  bit15                         bit               NULL,
  bit16                         bit               NULL,
  uniqueidentifier1             uniqueidentifier  NULL,
  tp_Level                      tinyint           NOT NULL   DEFAULT ((1)),
  tp_IsCurrentVersion           bit               NOT NULL   DEFAULT (CONVERT
                                                                 ([bit],(1),0)),
  tp_UIVersion                  int               NOT NULL   CONSTRAINT
                                [AllUserData_DEFAULT_UIVersionDEFAULT((512)),
  tp_CalculatedVersion          int               NOT NULL   CONSTRAINT
                                [AllUserData_DEFAULT_CalculatedVersionDEFAULT((0)),
  tp_UIVersionString            AS     ((CONVERT([nvarchar],[tp_UIVersion]/(512),0)+'.')
                                + CONVERT([nvarchar],[tp_UIVersion]%(512),0)),
  tp_DraftOwnerId               int               NULL       DEFAULT   (NULL),
  tp_CheckoutUserId             int               NULL       DEFAULT   (NULL)
  );
```

tp_Id: The identifier for the list item, uniquely identifying it within the **AllUserData** table.

tp_ListId: The List Identifier (section 2.2.1.5) of the list (1) or document library containing the list item.

tp_SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the list item.

tp_RowOrdinal: The zero-based ordinal index of this row in the set of rows representing the list item. Additional rows are used to represent list items that have more application-defined columns of one or more data types than can fit in a single row in the **AllUserData** table.

tp_Version: A counter incremented any time a change is made to the list item, used for internal conflict detection. Due to the mapping of application properties to the generic columns schema in this table, changes to application schema as well as property values can affect a version increment.

tp_Author: The User Identifier (section 2.2.1.13) for the user who created the list item.

tp_Editor: The User Identifier for the user who last edited the list item.

tp_Modified: A date and time value in UTC format specifying when this list item was last modified.

Copyright © 2014 Microsoft Corporation.

tp_Created: A date and time value in UTC format specifying when this list item was created.

tp_Ordering: A concatenation of time stamp values in yyyyMMddHHmmss format, specifying the threading structure of the list items in a legacy discussion board list (a list (1) with a List Base Type (section 2.2.3.11) of 3). For list items in all other types of list (1), this parameter MUST be NULL.

tp_ThreadIndex: A binary structure specifying the list item's position within a legacy discussion board list (1) (a list with a List Base Type of 3). For list items in all other types of list (1), this parameter MUST be NULL.

tp_HasAttachment: A bit set to 1 if the list item has an attachment associated with it; otherwise, it is set to 0.

tp_ModerationStatus: A Moderation Status (section 2.2.3.13) value indicating the current moderation approval status of the list item.

tp_IsCurrent: A bit set to 1 if this is a current version of this list item; otherwise, it is set to 0.

tp_ItemOrder: A value used to calculate the relative order in which to view the list item when displayed with other list items from the same list (1).

tp_InstanceID: If this list item is associated with a particular instance of a recurring meeting, this is the integer ID of that instance. For all other list items, this MUST be NULL.

tp_GUID: A List Item Identifier (section 2.2.1.6) value uniquely identifying this list item.

tp_CopySource: The URL used as a source for the list item. If this list item was not copied from a source list item, this value MUST be NULL.

tp_HasCopyDestinations: This bit is set to 1 if destination locations for the list item to be copied to have been set. If the list item does not have a destination location set, this value MUST be 0.

tp_AuditFlags: An Audit Flags (section 2.2.2.1) value determining the operations to be tracked on this list item.

tp_InheritAuditFlags: An Audit Flags value for the operations to be tracked on this list item, as determined from parent container Audit Flags settings.

tp_Size: The sum of the size, in bytes, of the content of application-schema columns in the list item. This does not include the size of the stream for list items that have an associated stream.

tp_WorkflowVersion: If the list item is part of a workflow (2), this stores an integer denoting the state of this list item within that workflow (2). Otherwise, this value MUST be NULL.

tp_WorkflowInstanceID: A Workflow Identifier (section 2.2.1.16) value for the currently active workflow instance on this list item. If the list item is not part of a workflow (2), this value MUST be NULL.

tp_DirName: The directory name of the list item.

tp_LeafName: The leaf name of the list item.

tp_DeleteTransactionId: An identifier for use with the implementation-specific deleted items recycle bin. This MUST equal 0x if the list item is nondeleted.

tp_ContentType: The user-friendly name of the content type associated with the list item.

tp_ContentTypeId: The binary identifier of the content type associated with the list item.

Copyright © 2014 Microsoft Corporation.

The next seven columns are duplicated a number of times within the table definition. This is indicated using a suffix '#', which is replaced with a numeral in the column names, described as follows. The number of times each column is duplicated varies and is indicated for each column. Each group of columns is dedicated to hold application-defined fields of a different data type, described as follows.

nvarchar#: Columns for application-defined fields that hold values of type nvarchar. The 64 columns are named nvarchar1 to nvarchar64. If the column does not contain data, this value MUST be NULL.

ntext#: Columns for application-defined fields that hold values of type ntext. The 32 columns are named ntext1 to ntext32. If the column does not contain data, this value MUST be NULL.

sql_variant#: Columns for application-defined fields that hold values of type sql_variant. The 8 columns are named sql_variant1 to sql_variant8. If the column does not contain data, this value MUST be NULL.

int#: Columns for application-defined fields that hold values of type int. The 16 columns are named int1 to int16. If the column does not contain data, this value MUST be NULL.

float#: Columns for application-defined fields that hold values of type float. The 12 columns are named float1 to float12. If the column does not contain data, this value MUST be NULL.

datetime#: Columns for application-defined fields that hold values of type datetime. The 8 columns are named datetime1 to datetime8. If the column does not contain data, this value MUST be NULL.

bit#: Columns for application-defined fields that hold values of type bit. The 16 columns are named bit1 to bit16. If the column does not contain data, this value MUST be NULL.

uniqueidentifier1: A column for an application-defined field (2) of type uniqueidentifier. If the column does not contain data, this value MUST be NULL.

tp_Level: A publishing level value specifying the publishing status of this version of the list item.

tp_IsCurrentVersion: A bit indicating whether this row corresponds to a current version or an historical version of the list item. This value MUST be 1 if this row contains a current version. Otherwise, it MUST be 0.

tp_UIVersion: The UI version number associated with this list item. The default value of **tp_UIVersion** is 512, which corresponds to a displayed version of 1.0.

tp_CalculatedVersion: This contains the UI version number if this is an historical version of the list item. This MUST be 0 if the list item is a current version.

tp_UIVersionString: A calculated column containing the value of the **tp_UIVersion** column as a displayed version string.

tp_DraftOwnerId: The identifier of the user who created this list item as a draft. This value is non-NULL only if the list item exists and is a draft version.

tp_CheckoutUserId: The identifier of the user who checked out this list item. This value is non-NULL only if the list item exists and is checked out.

2.2.7.4 Docs View

The Docs View reflects all rows in the table **AllDocs** (section 2.2.7.1) that are not deleted (that is, all table rows where DeleteTransactionId is equal to 0x). Docs View records include items that have

Copyright © 2014 Microsoft Corporation.

a URL, such as files, folders, lists, document libraries, and sites, referred to collectively as documents.

See AllDocs for the T-SQL syntax description of the structure of this view.

2.2.7.5 GroupMembership Table

The **GroupMembership** Table stores a list of the principals that are members of each site group in a site collection. The GroupMembership Table is defined using T-SQL syntax, as follows.

```
TABLE GroupMembership(
      SiteId                        uniqueidentifier   NOT NULL,
      GroupId                       int                NOT NULL,
      MemberId                      int                NOT NULL
);
```

SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the principals and site groups.

GroupId: The site group identifier of the site group that the principal (1) specified by **MemberId** is a member of.

MemberId: The User Identifier (section 2.2.1.13) of a principal (1) that is a member of the site group specified by **GroupId**.

2.2.7.6 Sec_SiteGroupsView

The **Sec_SiteGroupsView** is a view into site group information with one site group per row. Site groups available through this view are owned by a user or domain group, or by a site group. If a user or domain group owns the site group, the **OwnerIsUser** bit is set to 1, and the **GroupID**, **GroupName**, **GroupDescription**, **GroupSiteID**, **GroupOwner** and **GroupOwnerIsUser** fields are set to NULL.

If the site group is owned by a site group, the **OwnerIsUser** bit is set to 0, and the **UserID**, **UserSID**, **UserName**, **UserLogin**, **UserEmail**, **UserNotes**, **UserSiteAdmin** and UserDomainGroup are set to NULL. The **Sec_SiteGroupsView** Table is defined using T-SQL syntax, as follows.

```
VIEW Sec_SiteGroupsView(
      ID                                int,
      Title                             nvarchar(255),
      Description                       nvarchar(512),
      SiteWebId                         uniqueidentifier,
      Owner                             int,
      OwnerIsUser                       bit,
      OwnerGlobal                       bit,
      Type                              tinyint,
      Global                            bit,
      SiteID                            uniqueidentifier,
      UserID                            int,
      UserSID                           varbinary(512),
      UserName                          nvarchar(255),
      UserLogin                         nvarchar(255),
      UserEmail                         nvarchar(255),
      UserNotes                         nvarchar(1023),
      UserSiteAdmin                     bit,
```


Copyright © 2014 Microsoft Corporation.

```
        UserDomainGroup                 bit,
        GroupID                         int,
        GroupName                       nvarchar(255),
        GroupDescription                nvarchar(512),
        GroupSiteID                     uniqueidentifier,
        GroupOwner                      int,
        GroupOwnerIsUser                bit,
        DLAlias                         nvarchar(128),
        DLErrorMessage                  nvarchar(512),
        DLFlags                         int,
        DLJobId                         int,
        DLArchives                      varchar(4000),
        RequestEmail                    nvarchar(255),
        Flags                           int
);
```

ID: The identifier for the site group. This parameter MUST NOT be NULL.

Title: The user-friendly display name for the site group. This parameter MUST NOT be NULL.

Description: The site group description.

SiteWebId: The Site Collection Identifier (section 2.2.1.9) of the site collection that contains the site group information. This value is the same as returned in **SiteId**. For non-user-owned groups, this column name will be WebId. This parameter MUST NOT be NULL.

Owner: The User Identifier (section 2.2.1.13) or Site Group Identifier (section 2.2.1.10) of the site group owner. This parameter MUST NOT be NULL.

OwnerIsUser: A bit flag specifying whether the site group owner is a user or a site group. When the value in the **Owner** field (2) is a User Identifier for a user or a domain group, the **OwnerIsUser** flag MUST be set to 1. When the value in the Owner field (2) is a Site Group Identifier, the **OwnerIsUser** flag MUST be set to 0. This parameter MUST NOT be NULL.

OwnerGlobal: A bit flag specifying whether the view contains ownership information for a user or domain group, or for a site group.

When the owner is a site group, the **OwnerGlobal** flag MUST be set to 1, and the **GroupID**, **GroupName**, **GroupDescription**, **GroupSiteID**, **GroupOwner** and **GroupOwnerIsUser** fields MUST be populated with information from the site group owning this site group, and the **UserID**, **UserSID**, **UserName**, **UserLogin**, **UserEmail**, **UserNotes**, **UserSiteAdmin** and **UserDomainGroup** MUST be NULL.

When the owner is a user or domain group, the **OwnerGlobal** flag MUST be set to 0, the UserID, **UserSID**, **UserName**, **UserLogin**, **UserEmail**, **UserNotes**, **UserSiteAdmin** and **UserDomainGroup** MUST be populated with information from the user or domain group owning this site group, and the **GroupID**, **GroupName**, **GroupDescription**, **GroupSiteID**, **GroupOwner** and **GroupOwnerIsUser** MUST be NULL.

Type: This value MUST be 0.

Global: This value MUST be 1.

SiteID: The Site Collection Identifier of the site collection. This parameter MUST NOT be NULL.

UserID: The User Identifier of the owner of the site group. This MUST be NULL when the site group owner is a site group.

Copyright © 2014 Microsoft Corporation.

UserSID: The **SystemID** (section 2.2.1.12) of the owner of the site group. This MUST be NULL when the site group owner is a site group.

UserName: The user-friendly name of the owner of the site group. This MUST be NULL when the site group owner is a site group.

UserLogin: The login name of the owner of the site group. This MUST be NULL when the site group owner is a site group.

UserEmail: The email address of the owner of the site group. This MUST be NULL when the site group owner is a site group.

UserNotes: The notes associated with the owner of the site group. This MUST be NULL when the site group owner is a site group.

UserSiteAdmin: A bit flag specifying whether the site group owner is a site administrator. When the site group owner is a site administrator, the **UserSiteAdmin** flag MUST be set to 1. If the user or domain group owner of the site group is not a site administrator, the **UserSiteAdmin** flag MUST be set to 0. This MUST be NULL when when the site group owner is a site group.

UserDomainGroup: A bit flag specifying whether the site group owner is a domain group. When the site group owner is a domain group, the flag MUST be set to 1. When the site group owner is a user or domain group, the flag MUST be set to 0. This flag MUST be NULL when the site group owner is a site group.

GroupID: The Site Group Identifier of the owner of this site group. This MUST be NULL when the site group owner is a user or domain group.

GroupName: The user-friendly name of the owner of this site group. This MUST be NULL when the site group owner is a user or domain group.

GroupDescription: The description of the owner of this site group. This MUST be NULL when the site group owner is a user or domain group.

GroupSiteID: The Site Collection Identifier of the site collection containing the owner of this site group. This MUST be NULL when the site group owner is a user or domain group.

GroupOwner: The User Identifier or Site Group Identifier of the owner of the site group that owns this site group. This MUST be NULL when the site group owner is a user or domain group, because only an owner that is a site group can have a group owner.

GroupOwnerIsUser: A bit flag specifying whether the site group that owns this site group is in turn owned by a user or domain group, or by a site group. When the owner of the site group that owns this site group is a user or domain group, the **GroupOwnerIsUser** flag MUST be set to 1. This flag MUST be set to 0 when the site group owner is owned by a site group. This MUST be NULL when the site group owner is a user or domain group.

DLAlias**:** The email distribution list address for the site group. This value MUST be NULL if the group has no email distribution list address.

DLErrorMessage: Contains the most recent error message returned by an asynchronous email distribution list operation, if any.

DLFlags: Contains a bit field (2) of status flags for the email distribution list associated with this site group. This parameter MUST be one of the values listed in the following table.

Copyright © 2014 Microsoft Corporation.

Value	Description
0x00000000	None.
0x00000001	Pending – If this bit is set, the email distribution list associated with this site group currently has a pending asynchronous operation.
0x00000002	Dirty – If this bit is set, the email distribution list associated with this group is dirty. The email distribution list is considered dirty when A) an asynchronous operation is pending, and B) the site group has been modified since the asynchronous operation was initiated. A dirty email distribution list needs to have its status and membership synchronized as soon as the pending operation completes (whether it succeeds or fails).
0x00000004	PendingJobIsRename - The type of the currently pending asynchronous email distribution list operation, if any. When this bit is set, the pending operation is a Rename. When the bit is not set, it is a Create operation.

DLJobId: Contains the job ID of the currently pending asynchronous email distribution list operation or 0 if there is no pending operation.

DLArchives: An array of bytes containing a list of pairs of List Identifiers (section 2.2.1.5) defining additional lists, which are recipients of email sent to the email distribution list via the address in DLAlias. Each Identifier MUST be stored as a string, with commas separating the list's parent site document identifier (section 2.2.1.2) and the List Identifier, and with semicolons following each pair.

RequestEmail: The email address used to send a request to join or depart a site group.

Flags: Contains the membership permissions bit field (2) for the site group. This parameter MUST NOT be NULL and MUST be one of the values listed in the following table.

Value	Description
0x00000000	Allow anyone to view the membership of the site group.
0x00000001	Only allow members of the site group to view the membership.
0x00000002	Allow members of the site group to edit the membership of the site group.
0x00000004	Allow users to request membership in this site group, and allow users to request to leave the site group. All requests MUST be sent to the email address specified by RequestEmail.
0x00000008	Automatically accept user requests to join or leave the site group. This bit MUST be set only when the bit 0x00000004 is also set.

2.2.7.7 Sites Table

The Sites Table stores information about site collections. The Table is defined using T-SQL syntax, as follows.

```
TABLE <table name>(
      FullUrl                       nvarchar(255) DEFAULT N'',
      Id                            uniqueidentifier,
      NextUserOrGroupId             int,
      OwnerID                       int,
      SecondaryContactID            int,
      Subscribed                    bit,
```


Copyright © 2014 Microsoft Corporation.

```
        TimeCreated                 datetime,
        UsersCount                  int           DEFAULT ((1)),
        BWUsed                      bigint,
        DiskUsed                    bigint,
        SecondStageDiskUsed         bigint,
        QuotaTemplateID             smallint,
        DiskQuota                   bigint,
        UserQuota                   int,
        DiskWarning                 bigint,
        DiskWarned                  datetime,
        BitFlags                    int,
        SecurityVersion             bigint,
        CertificationDate           datetime,
        DeadWebNotifyCount          smallint,
        PortalURL                   nvarchar(260),
        PortalName                  nvarchar(255),
        LastContentChange           datetime      DEFAULT (getutcdate()),
        LastSecurityChange          datetime      DEFAULT (getutcdate()),
        AuditFlags                  int,
        InheritAuditFlags           int,
        UserInfoListId              uniqueidentifier,
        UserIsActiveFieldRowOrdinal int,
        UserIsActiveFieldColumnName nvarchar(64),
        UserAccountDirectoryPath    nvarchar(255),
        RootWebId                   uniqueidentifier,
        HashKey                     binary(16),
        DomainGroupMapVersion       bigint,
        DomainGroupMapCacheVersion  bigint        DEFAULT ((-1)),
        DomainGroupMapCache         image,
        HostHeader                  nvarchar(128),
        EmailEnabled                bit
);
```

FullUrl: The absolute URL of the site collection.

Id: The Site Collection Identifier (section 2.2.1.9) of the site collection. Used only during upgrade.

NextUserOrGroupId: An integer value that is incremented when a new user or site group is added to the site collection. Indicates that the value of the next user or Site Group Identifier (section 2.2.1.10) to be used.

OwnerID: The User Identifier (section 2.2.1.13) of the user who owns the site collection.

SecondaryContactID: The User Identifier of the user who is the secondary contact of the site collection.

Subscribed: A bit set to 1 to indicate that the site collection has been subscribed for implementation-specific notifications.

TimeCreated: The date and time in UTC format when the site collection was created.

UsersCount: The number of users in the site collection.

BWUsed: The number of sites in the site collection actively used. Serves a Windows SharePoint Services implementation-specific, usage-reporting feature.

DiskUsed: The size of disk space used to store content in the site collection, in bytes.

Copyright © 2014 Microsoft Corporation.

SecondStageDiskUsed: The size of disk space used to store the second-stage trash bin items for the site collection, in bytes.

QuotaTemplateID: An identifier of a quota template used to set the disk quota for the site collection.

DiskQuota: The maximum size, in bytes, of disk space that can be allocated by the site collection. A value of 0 indicates that no limit is set.

UserQuota: The maximum number of users that the site collection can contain. A value of 0 indicates that no limit is set.

DiskWarning: An integer value indicating the number of times a warning was sent to the site collection owner about disk usage.

DiskWarned: A date and time value in UTC format set to the last time a warning was sent to the site collection owner about disk usage, if any.

BitFlags: A Site Collection Flags (section 2.2.2.9) value describing the site collection.

SecurityVersion: A version number incremented when changes are made to the site collection's permissions.

CertificationDate: The date and time, in UTC format, when the site collection was last confirmed by its owner as being used.

DeadWebNotifyCount: The number of times that a notification was sent to the site collection owner that the site collection will be deleted if the owner does not certify it as being used. (See **CertificationDate**.)

PortalURL: The URL of an external website designated as the sites' portal. This is a Windows SharePoint Services implementation-specific navigation links feature.

PortalName: The name of the external website referenced in the **PortalURL** column. Used for display in the Windows SharePoint Services implementation-specific navigation feature.

LastContentChange: The date and time value, in UTC format, that the content of the site collection was last changed.

LastSecurityChange: The date and time value, in UTC format, when the permissions on the site collection were last changed.

AuditFlags: An Audit Flags (section 2.2.2.1) value specifying the operations to be audited that are set directly on the site collection.

InheritAuditFlags: An Audit Flags value specifying the operations to be audited on the specified object that are inherited.

UserInfoListId: The List Identifier (section 2.2.1.5) of a list holding user information for the site collection.

UserIsActiveFieldRowOrdinal: The ordinal of the column in the list identified by UserInfoList Id that tracks whether a user is an active user in the site collection.

UserIsActiveFieldColumnName: The name of the column in the list identified by UserInfoListId that tracks whether a user is an active user in the site collection.

Copyright © 2014 Microsoft Corporation.

UserAccountDirectoryPath: A provider-specific user account path. The value is used for validation of added user accounts if the site collection flags bit 0x00080000 is set in the BitFlags column.

RootWebId: The Site Identifier (section 2.2.1.11) of the site configured as the root site (2) in the site collection.

HashKey: A hash key associated with the site collection.

DomainGroupMapVersion: The version of the domain group map.

DomainGroupMapCacheVersion: The version of the domain group map cache.

DomainGroupMapCache: A binary serialization of the domain group map cache, containing a mapping of external groups to the site groups of which they are members.

HostHeader: When used in host header mode, this contains the Host Header string associated with the site collection's web application (1).

EmailEnabled: A bit indicating if the site collection has Windows SharePoint Services implementation-specific email integration features enabled. Set to 1 if enabled; otherwise, set to 0.

2.2.7.8 UserData View

The **UserData View** reflects all list items in the table AllUserData (section 2.2.7.3) that are current versions and not marked as deleted (that is, all table rows where **tp_IsCurrentVersion** is 1, **tp_CalculatedVersion** is 0, and **tp_DeleteTransactionId** is 0x).

See **AllUserData** for the T-SQL syntax description of the structure of this view.

2.2.7.9 UserDataVersioned View

The UserDataVersioned View reflects all list items in the table AllUserData (section 2.2.7.3) that are not deleted (that is, all table rows where **tp_DeleteTransactionId** is 0x).

See **AllUserData** for the T-SQL syntax description of the structure of this view.

2.2.7.10 UserInfo Table

The UserInfo Table stores descriptive properties and security information about principals with access to a site collection. The UserInfo Table is defined using T-SQL syntax, as follows.

```
TABLE UserInfo (
tp_SiteID                       uniqueidentifier    NOT NULL,
tp_Id                           int                 NOT NULL,
tp_DomainGroup                  bit                 NOT NULL,
tp_GUID                         uniqueidentifier    NOT NULL DEFAULT (newid()),
tp_SystemID                     varbinary(512)      NOT NULL,
tp_Deleted                      int                 NOT NULL,
tp_SiteAdmin                    bit                 NOT NULL,
tp_IsActive                     bit                 NOT NULL DEFAULT (1),
tp_Login                        nvarchar(255)       NOT NULL,
tp_Title                        nvarchar(255)       NOT NULL,
tp_Email                        nvarchar(255)       NOT NULL,
tp_Notes                        nvarchar(1023)      NOT NULL,
tp_Token                        image               NULL,
tp_ExternalToken                image               NULL,
tp_ExternalTokenLastUpdated     datetime            NULL,
```


Copyright © 2014 Microsoft Corporation.

```
tp_Locale                       int               NULL,
tp_CalendarType                 smallint          NULL,
tp_AdjustHijriDays              smallint          NULL,
tp_TimeZone                     smallint          NULL,
tp_Time24                       bit               NULL,
tp_AltCalendarType              uniqueidentifier  NULL,
tp_CalendarViewOptions          uniqueidentifier  NULL,
tp_WorkDays                     smallint          NULL,
tp_WorkDayStartHour             smallint          NULL,
tp_WorkDayEndHour               smallint          NULL
);
```

tp_SiteID: The Site Collection Identifier (section 2.2.1.9) of the site collection associated with the principal (1).

tp_Id: The User Identifier (section 2.2.1.13) of the principal (1), which MUST be unique within both this table and the group identifiers for site groups. Certain User Identifiers are predefined, such as the system account (predefined as 1073741823), the site collection owner (predefined as 1), and the secondary site collection contact (predefined as 2). User Identifier values of 0 and -1 MUST NOT be used in the table.

tp_DomainGroup: A bit set to 1 if the principal (1) is a domain group; otherwise, 0, specifying that the principal is a user.

tp_GUID: A GUID for indexing with the UserInfo table.

tp_SystemID: The **SystemID** (section 2.2.1.12) of the principal (1).

tp_Deleted: A value specifying whether the principal (1) is marked as deleted. If **tp_**Deleted is 0, the principal is not deleted. If **tp_**Deleted is not 0, the principal (1) is marked as deleted, and the value is the User Identifier that was associated with this principal. The deleted state can be used for user information, rather than dropping entries from the table, to preserve list item ownership information. A user or domain group with the **tp_**Deleted value set to nonzero can be restored by setting the **tp_Deleted** value to 0 and updating other fields as necessary.

tp_SiteAdmin: A bit set to 1 if the principal (1) is a site administrator for the site collection.

tp_IsActive: A bit set to 1 if the principal (1) is an active user in the site collection.

tp_Login: The login name for the principal (1). This value MUST NOT be empty.

tp_Title: The display name for the principal (1). This value MUST NOT be empty.

tp_Email: The email address for the principal (1). This value can be empty.

tp_Notes: A descriptive text about the principal (1). This value can be empty.

tp_Token: A WSS User Token (section 2.2.4.9) value specifying the site group membership of the principal (1). This can be NULL, indicating that the principal (1) is not a member of a site group.

tp_ExternalToken: An External Group Token (section 2.2.4.2) value encoding information on external group membership derived from an external authentication role provider. 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.

Copyright © 2014 Microsoft Corporation.

tp_ExternalTokenLastUpdated: A datetime containing a time stamp in UTC format specifying the time when the value in **tp_ExternalToken** for the principal (1) was last updated. This can be NULL if the value in **tp_ExternalToken** has never been updated.

tp_Locale: An LCID specifying the preferred locale value to be used when displaying messages in the WFE for the principal (1). This can be NULL, specifying that the system or site (2) default locale can be used instead.

tp_CalendarType: The Calendar Type to be used when processing date values for the principal (1). This can be NULL, specifying that the site (2) default Calendar Type can be used instead.

tp_AdjustHijriDays: If the **tp_**CalendarType value is "6", this specifies the number of days to extend or reduce the current month in Hijri calendars for the principal (1). Otherwise, this value MUST be NULL and MUST be ignored.

tp_TimeZone: The Time Zone Identifier for the time zone to be used when displaying time values for the principal (1). This can be NULL, specifying that the system or site (2) default time zone can be used instead.

tp_Time24: A bit flag specifying whether a 24-hour time format can be used when displaying time values to the principal (1). If **tp_**Time24 is 1, the 24-hour time format will be used; otherwise, the 12-hour time format will be used. This can be NULL, specifying that the system or site (2) default can be used instead.

tp_AltCalendarType: The Calendar Type of an alternate calendar to be used when displaying calendars in views for the principal (1). This can be NULL, specifying that the **tp_**CalendarType value can be used instead.

tp_CalendarViewOptions: A Calendar View Options Type that specifies the calendar options for the principal (1). This can be NULL, specifying that the site (2) default calendar view options can be used instead.

tp_WorkDays: A set of Workdays Flags that specify the weekdays defined as the workweek for the principal (1). This can be NULL, specifying that the site (2) default Workdays value can be used instead.

tp_WorkDayStartHour: The start time of the workday in minutes after midnight for the principal (1). This can be NULL, specifying that the site (2) default workday start hour can be used instead.

tp_WorkDayEndHour: The end time of the workday in minutes after midnight for the principal (1). This can be NULL, specifying that the site (2) default workday end hour can be used instead.

2.2.7.11 Versions

The Versions Table stores information used for object versioning during upgrade operations. The Table is defined using T-SQL syntax, as follows.

```
TABLE Versions(
      VersionId                   uniqueidentifier NOT NULL,
      Version                     nvarchar(64) NOT NULL,
      Id                          int IDENTITY(1,1) NOT NULL,
      UserName                    nvarchar(255) NULL,
      TimeStamp                   datetime NULL,
      FinalizeTimeStamp           datetime NULL,
      Mode                        int NULL,
      ModeStack                   int NULL,
      Updates                     int NOT NULL DEFAULT((0)),
```


Copyright © 2014 Microsoft Corporation.

```
    Notes                          nvarchar (1024) NULL
);
```

VersionId: A GUID that identifies the version.

Version: The actual version string in the format <major>.<minor>.<build>.<iteration> (for example, "12.0.0.4518").

Id: An integer value that is an SQL identity column.

UserName: A string containing the user name adding or updating the version.

TimeStamp: A time stamp in UTC format indicating when the version was added or updated.

FinalizeTimeStamp: A time stamp in UTC format indicating when the upgrade to the version was completed.

Mode: An integer value indicating the type of upgrade used. The following values are valid.

Value	Description
0	In-place upgrade: Upgrading an existing instance in place.
1	Side-by-side upgrade: Migrating an instance to another database.
2147483647	(0x7FFFFFFF) None. Not currently being upgraded or migrated.

ModeStack: A list of **Mode** values used in sequence to reach the current mode, as when different database types are upgraded in a nested fashion. For example, a farm might upgrade gradually, while a user executes an in-place upgrade on a subset of the data. **ModeStack** indicates the mode of the child upgrade operation.

Updates: An integer value tracking the number of updates applied to a particular version. This value MUST be initialized to 1 when inserted and incremented each time the version is updated.

Notes: A string containing optional notes associated with the version.

2.2.8 XML Structures

2.2.8.1 Namespaces

None.

2.2.8.2 Simple Types

2.2.8.2.1 FALSE_Case_Insensitive_Else_Anything

The **FALSE_Case_Insensitive_Else_Anything** type is used to specify a Boolean value.

```
<xs:simpleType name="FALSE_Case_Insensitive_Else_Anything">
  <xs:restriction base="xs:string">
    <xs:pattern value="[Ff][Aa][Ll][Ss][Ee]|.*" />
  </xs:restriction>
</xs:simpleType>
```


Copyright © 2014 Microsoft Corporation.

2.2.8.2.2 FieldAggregationAttribute

The **FieldAggregationAttribute** type is used to specify how values are promoted from an XML or site template document.

```
<xs:simpleType name="FieldAggregationAttribute">
    <xs:restriction base="xs:string">
        <xs:enumeration value="average" />
        <xs:enumeration value="count" />
        <xs:enumeration value="first" />
        <xs:enumeration value="last" />
        <xs:enumeration value="max" />
        <xs:enumeration value="merge" />
        <xs:enumeration value="min" />
        <xs:enumeration value="plaintext" />
        <xs:enumeration value="signature" />
        <xs:enumeration value="sum" />
    </xs:restriction>
</xs:simpleType>
```

The meanings of the values are specified in the following table. In the table, a matching node designates an element or an attribute that matches a specified XPath query. The value of an element node is the concatenation of all character data directly contained by the element.

Value	Meaning
average	The average of the matching nodes, whose values can be interpreted as a floating point numbers. This is an empty string if no values match.
count	Count of the matching nodes.
first	The value of the first matching node.
last	The value of the last matching node.
max	The value of the largest matching node interpreted as a floating point number.
merge	The value of the nodes, in order, delimited by a newline character.
min	The value of the smallest matching node interpreted as a floating point number.
plaintext	If no nodes match, or if the first matching node is not an element, then this is an empty string. Otherwise, it renders the first matching node and its child elements with only the character data shown.
signature	TRUE if the first matching node is an element and has child elements. FALSE if the first matching node is an element with no child elements. Otherwise, an empty string.
sum	The sum of the nodes that match and whose values can be interpreted as floating point numbers.

Copyright © 2014 Microsoft Corporation.

2.2.8.2.3 FieldInternalType

The **FieldInternalType** type is used to specify a field (2) internal type. Note In the following XML, the value attribute is broken into multiple lines only for readability.

```
  <xs:simpleType name="FieldInternalType">
    <xs:restriction base="xs:string">
      <xs:pattern value="[iI][nN][tT][eE][gG][eE][rR]|[tT][eE][xX]
[tT]|[nN][oO][tT][eE]|[dD][aA][tT][eE][tT][iI][mM][eE]|[cC][oO][uU][nN]
[tT][eE][rR]|[cC][hH][oO][iI][cC][eE]|[lL][oO][oO][kK][uU][pP]|[bB][oO]
[oO][lL][eE][aA][nN]|[nN][uU][mM][bB][eE][rR]|[cC][uU][rR][rR][eE][nN]
[cC][yY]|[uU][rR][lL]|[cC][oO][mM][pP][uU][tT][eE][dD]|[tT][hH][rR][eE]
[aA][dD][iI][nN][gG]|[gG][uU][iI][dD]|[mM][uU][lL][tT][iI][cC][hH][oO]
[iI][cC][eE]|[gG][rR][iI][dD][cC][hH][oO][iI][cC][eE]|[cC][aA][lL][cC]
[uU][lL][aA][tT][eE][dD]|[fF][iI][lL][eE]|[aA][tT][tT][aA][cC][hH][mM]
[eE][nN][tT][sS]|[uU][sS][eE][rR]|[rR][eE][cC][uU][rR][rR][eE][nN][cC]
[eE]|[cC][rR][oO][sS][sS][pP][rR][oO][jJ][eE][cC][tT][lL][iI][nN][kK]|
[mM][oO][dD][sS][tT][aA][tT]|[eE][rR][rR][oO][rR]|[cC][oO][nN][tT][eE]
[nN][tT][tT][yY][pP][eE][iI][dD]|[pP][aA][gG][eE][sS][eE][pP][aA][rR]
[aA][tT][oO][rR]|[tT][hH][rR][eE][aA][dD][iI][nN][dD][eE][xX]|[wW][oO]
[rR][kK][fF][lL][oO][wW][sS][tT][aA][tT][uU][sS]|[aA][lL][lL][dD][aA]
[yY][eE][vV][eE][nN][tT]|[wW][oO][rR][kK][fF][lL][oO][wW][eE][vV][eE]
[nN][tT][tT][yY][pP][eE]" />
    </xs:restriction>
  </xs:simpleType>
```

While the XSD pattern is normative, the values are:

- Integer
- Text
- Note
- DateTime
- Counter
- Choice
- Lookup
- Boolean
- Number
- Currency
- Url
- Computed
- Threading
- GUID
- MultiChoice
- GridChoice

Copyright © 2014 Microsoft Corporation.

- Calculated
- File
- Attachments
- User
- Recurrence
- CrossProjectLink
- ModStat
- Error
- ContentTypeId
- PageSeparator
- ThreadIndex
- WorkflowStatus
- AllDayEvent
- WorkFlowEventType

2.2.8.2.4 FieldRefType

The **FieldRefType** type is used to specify the type of relationship given by a field (2) reference definition within the field.

```
<xs:simpleType name="FieldRefType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Recurrence" />
    <xs:enumeration value="EventType" />
    <xs:enumeration value="UID" />
    <xs:enumeration value="RecurrenceId" />
    <xs:enumeration value="EventCancel" />
    <xs:enumeration value="StartDate" />
    <xs:enumeration value="EndDate" />
    <xs:enumeration value="RecurData" />
    <xs:enumeration value="Duration" />
    <xs:enumeration value="TimeZone" />
    <xs:enumeration value="XMLTZone" />
    <xs:enumeration value="CPLink" />
    <xs:enumeration value="LinkURL" />
    <xs:enumeration value="MasterSeriesItemID" />
    <xs:enumeration value="AllDayEvent" />
  </xs:restriction>
</xs:simpleType>
```

2.2.8.2.5 FieldRichTextMode

The **FieldRichTextMode** type is used to specify a formatting mode for a rich text field (2).

Copyright © 2014 Microsoft Corporation.

```
<xs:simpleType name="FieldRichTextMode">
  <xs:restriction base="xs:string">
      <xs:enumeration value="Compatible"/>
      <xs:enumeration value="FullHtml"/>
      <xs:enumeration value="HtmlAsXml"/>
  </xs:restriction>
</xs:simpleType>
```

The meanings of the values are specified in the following table.

Value	Meaning
Compatible	Specifies support for basic formatting, such as bold formatting.
FullHtml	Specifies support for more advanced formatting options, such as pictures, tables, and hyperlinks.
HtmlAsXml	Specifies support for HTML that is well-formed XML.

2.2.8.2.6 IMEMode

The **IMEMode** type is used to specify a bias for an Input Method Editor (IME).

```
<xs:simpleType name="IMEMode">
  <xs:restriction base="xs:string">
    <xs:enumeration value="inactive" />
    <xs:enumeration value="auto" />
    <xs:enumeration value="active" />
    <xs:enumeration value="disabled" />
  </xs:restriction>
</xs:simpleType>
```

The meanings of the values are specified in the following table.

Value	Meaning
inactive	All characters are entered without using the IME mode, but it can still be activated. For datetime type fields, the default is inactive for detailed forms.
auto	Default. The IME mode is not affected with this choice.
active	All characters are entered in the IME mode, but it can still be deactivated. For name-related fields and addresses, the default is active for detailed forms.
disabled	The IME mode is disabled, and it cannot be activated. For number type fields (money, int, float), the default is inactive for detailed forms.

The reader MUST accept any value. The writer SHOULD write this attribute as one of the valid values for the IME-mode attribute to be applied to an `<input>` tag. See [HTML], section 17.4.

2.2.8.2.7 IntPositive

The **IntPositive** type is used to specify a positive integer.

Copyright © 2014 Microsoft Corporation.

```
<xs:simpleType name="IntPositive">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="1" />
  </xs:restriction>
</xs:simpleType>
```

2.2.8.2.8 Join Type

The **JoinType** type specifies how to handle lookup fields for which the target of the lookup does not exist.

```
<xs:simpleType name="JoinType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="INNER" />
    <xs:enumeration value="LEFT OUTER" />
  </xs:restriction>
</xs:simpleType>
```

The meanings of the values are specified in the following table.

Value	Meaning
INNER	Items whose target does not exist are omitted.
LEFT OUTER	Items whose target does not exist are included.

2.2.8.2.9 TextDirection

The **TextDirection** type is used to specify the preferred direction of displaying text.

```
<xs:simpleType name="TextDirection">
  <xs:restriction base="xs:string">
    <xs:enumeration value="ltr" />
    <xs:enumeration value="rtl" />
    <xs:enumeration value="none" />
    <xs:enumeration value="None" />
  </xs:restriction>
</xs:simpleType>
```

The meanings of the values are specified in the following table.

Value	Meaning
ltr	Specifies a left-to-right text direction.
rtl	Specifies a right-to-left text direction.
none	Specifies no hint for a text direction, and that the direction will follow the context of the page.
None	Same as none.

Copyright © 2014 Microsoft Corporation.

2.2.8.2.10 TRUE_If_Present

The TRUE_If_Present type is used to specify a Boolean value.

```
<xs:simpleType name="TRUE_If_Present">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
```

2.2.8.2.11 TRUEFALSE

The **TRUEFALSE** type is used to specify a Boolean value.

```
<xs:simpleType name="TRUEFALSE">
  <xs:restriction base="xs:string">
    <xs:pattern value="[Tt][Rr][Uu][Ee]|[Ff][Aa][Ll][Ss][Ee]"/>
  </xs:restriction>
</xs:simpleType>
```

2.2.8.2.12 UniqueIdentifierWithoutBraces

The **UniqueIdentifierWithoutBraces** type is used to specify a GUID.

```
<xs:simpleType name="UniqueIdentifierWithoutBraces">
  <xs:restriction base="UniqueIdentifierWithoutBracesOrEmpty">
    <xs:minLength value="1"/>
  </xs:restriction>
</xs:simpleType>
```

2.2.8.2.13 UniqueIdentifierWithoutBracesOrEmpty

The **UniqueIdentifierWithoutBracesOrEmpty** type is used to specify a GUID.

```
  <xs:simpleType name="UniqueIdentifierWithoutBracesOrEmpty">
    <xs:restriction base="xs:string">
      <xs:pattern value="[0-9a-fA-F]{8}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-
9a-fA-F]{12}|"/>
    </xs:restriction>
  </xs:simpleType>
```

2.2.8.2.14 UniqueIdentifierWithOrWithoutBraces

The **UniqueIdentifierWithOrWithoutBraces** type is used to specify a GUID.

```
  <xs:simpleType name="UniqueIdentifierWithOrWithoutBraces">
    <xs:restriction base="xs:string">
      <xs:pattern value="\{?[0-9a-fA-F]{8}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-
[0-9a-fA-F]{12}\}?"/>
    </xs:restriction>
```


Copyright © 2014 Microsoft Corporation.

```
</xs:simpleType>
```

2.2.8.3 Complex Types

2.2.8.3.1 CHOICEDEFINITION Type

The **CHOICEDEFINITION** type contains a collection of choices for a **Choice** or **MultiChoice** field (2).

2.2.8.3.1.1 Schema

```
<xs:complexType name="CHOICEDEFINITION" mixed="true">
    <xs:attribute name="JumpTo" type="xs:string" />
</xs:complexType>
```

2.2.8.3.1.2 Attributes

JumpTo: Specifies the Name of the next field (2) in a survey list to display when this choice is selected.

2.2.8.3.1.3 Child Elements

<Content>: Specifies values for choices in a **Choice** or **MultiChoice** field (2).

2.2.8.3.2 CHOICEDEFINITIONS Type

The **CHOICEDEFINITIONS** type contains a collection of choices for a **Choice** or **MultiChoice** field (2).

2.2.8.3.2.1 Schema

```
  <xs:complexType name="CHOICEDEFINITIONS">
    <xs:sequence>
      <xs:element name="CHOICE" type="CHOICEDEFINITION" minOccurs="0" maxOccurs="unbounded"
/>
    </xs:sequence>
  </xs:complexType>
```

2.2.8.3.2.2 Attributes

None.

2.2.8.3.2.3 Child Elements

CHOICE: Specifies a choice in a **Choice** or **MultiChoice** field (2).

Copyright © 2014 Microsoft Corporation.

2.2.8.3.3 FieldDefinition Type

A **FieldDefinition** describes the structure and format of a field (2) that is used within a list (1) or content type.

2.2.8.3.3.1 Schema

```
  <xs:complexType name="FieldDefinition" mixed="true">
    <xs:all>
      <xs:element name="CHOICES" type="CHOICEDEFINITIONS" minOccurs="0" maxOccurs="1" />
      <xs:element name="Customization" minOccurs="0" maxOccurs="1">
          <xs:complexType>
              <xs:sequence>
                  <xs:any minOccurs="1" maxOccurs="1" namespace="##any"
processContents="skip" />
              </xs:sequence>
          </xs:complexType>
      </xs:element>
      <xs:element name="Default" type="xs:string" minOccurs="0" maxOccurs="1" />
      <xs:element name="DefaultFormula" type="xs:string" minOccurs="0" maxOccurs="1" />
      <xs:element name="DisplayBidiPattern" minOccurs="0" maxOccurs="1">
          <xs:complexType>
              <xs:sequence>
                  <xs:any minOccurs="0" maxOccurs="unbounded" namespace="##any"
processContents="skip" />
              </xs:sequence>
              <xs:anyAttribute processContents="skip" />
          </xs:complexType>
      </xs:element>
      <xs:element name="DisplayPattern" type="CamlViewRoot" minOccurs="0" maxOccurs="1">
          <xs:complexType>
              <xs:sequence>
                  <xs:any minOccurs="0" maxOccurs="unbounded" namespace="##any"
processContents="skip" />
              </xs:sequence>
              <xs:anyAttribute processContents="skip" />
          </xs:complexType>
      </xs:element>
      <xs:element name="FieldRefs" minOccurs="0" maxOccurs="1">
        <xs:complexType mixed="true">
          <xs:sequence>
            <xs:element name="FieldRef" type="FieldRefDefinitionField" minOccurs="0"
maxOccurs="unbounded" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="Formula" type="xs:string" minOccurs="0" maxOccurs="1" />
      <xs:element name="MAPPINGS" type="MAPPINGDEFINITIONS" minOccurs="0" maxOccurs="1" />
      <xs:element name="ParserRefs" type="FieldParserRefs" minOccurs="0" maxOccurs="1" />
    </xs:all>
    <xs:attribute name="Aggregation" type="FieldAggregationAttribute" default="first" />
    <xs:attribute name="aggregation" type="xs:string" />
    <xs:attribute name="AllowDeletion" type="TRUEFALSE" default="TRUE" />
    <xs:attribute name="AllowHyperlink" type="TRUEFALSE" default="FALSE" />
    <xs:attribute name="AllowMultiVote" type="TRUEFALSE" default="FALSE" />
    <xs:attribute name="AppendOnly" type="TRUEFALSE" default="FALSE" />
    <xs:attribute name="AuthoringInfo" type="xs:string" default=""/>
    <xs:attribute name="BaseRenderingType" type="FieldInternalType" />
    <xs:attribute name="BaseType" type="FieldInternalType" default="Text" />
```


Copyright © 2014 Microsoft Corporation.

```
<xs:attribute name="Calculated" type="xs:string" />
<xs:attribute name="CalType" type="xs:int" />
<xs:attribute name="CalendarType" type="xs:int" />
<xs:attribute name="CanToggleHidden" type="TRUEFALSE" default="FALSE" />
<xs:attribute name="CountRelated" type="TRUE_If_Present" default="FALSE" />
<xs:attribute name="ClassInfo" type="xs:string" default="" />
<xs:attribute name="ColName" type="xs:string" />
<xs:attribute name="ColName2" type="xs:string" />
<xs:attribute name="Commas" type="TRUEFALSE" />
<xs:attribute name="Customization" type="xs:string" />
<xs:attribute name="Decimals" type="xs:int" default="-1"/>
<xs:attribute name="DefaultURLDesc" type="xs:string" />
<xs:attribute name="Description" type="xs:string" />
<xs:attribute name="Direction" type="TextDirection" default="none" />
<xs:attribute name="Dir" type="xs:string" />
<xs:attribute name="DisplaceOnUpgrade" type="TRUEFALSE" default="FALSE" />
<xs:attribute name="DisplayImage" type="xs:string" />
<xs:attribute name="DisplayName" type="xs:string" />
<xs:attribute name="DisplayNameSrcField" type="xs:string" />
<xs:attribute name="DisplaySize" type="xs:int" />
<xs:attribute name="Div" type="xs:string" default="1.0" />
<xs:attribute name="EnableLookup" type="TRUEFALSE" default="FALSE" />
<xs:attribute name="ExceptionImage" type="xs:string" />
<xs:attribute name="Ext" type="TRUEFALSE" default="FALSE" />
<xs:attribute name="FieldRef" type="xs:string" />
<xs:attribute name="FillInChoice" type="TRUEFALSE" default="FALSE" />
<xs:attribute name="Filterable" type="TRUEFALSE" default="TRUE" />
<xs:attribute name="FilterableNoRecurrence" type="TRUEFALSE" default="FALSE" />
<xs:attribute name="ForcedDisplay" type="xs:string" />
<xs:attribute name="Format" type="xs:string" />
<xs:attribute name="FromBaseType" type="TRUEFALSE" default="FALSE" />
<xs:attribute name="GridEndNum" type="xs:int" />
<xs:attribute name="GridNATxt" type="xs:string" default="" />
<xs:attribute name="GridStartNum" type="IntPositive" />
<xs:attribute name="GridTxtRng1" type="xs:string" default="" />
<xs:attribute name="GridTxtRng2" type="xs:string" default="" />
<xs:attribute name="GridTxtRng3" type="xs:string" default="" />
<xs:attribute name="Group" type="xs:string" />
<xs:attribute name="HeaderImage" type="xs:string" />
<xs:attribute name="Height" type="xs:int" />
<xs:attribute name="Hidden" type="TRUEFALSE" default="FALSE" />
<xs:attribute name="ID" type="UniqueIdentifierWithOrWithoutBraces" />
<xs:attribute name="Id" type="xs:string" />
<xs:attribute name="IMEMode" type="IMEMode" />
<xs:attribute name="Indexed" type="TRUEFALSE" default="FALSE" />
<xs:attribute name="ImnHeader" type="xs:string" />
<xs:attribute name="IsolateStyles" type="TRUEFALSE" default="FALSE" />
<xs:attribute name="JoinColName" type="xs:string" default="tp_ID" />
<xs:attribute name="JoinRowOrdinal" type="xs:int" fixed="0" />
<xs:attribute name="JoinType" type="JoinType" default="LEFT OUTER" />
<xs:attribute name="JumpTo" type="xs:string" />
<xs:attribute name="JumpToFillinChoice" type="xs:string" />
<xs:attribute name="JumpToNo" type="xs:string" />
<xs:attribute name="JumpToYes" type="xs:string" />
<xs:attribute name="LCID" type="xs:int" />
<xs:attribute name="List" type="xs:string" />
<xs:attribute name="Max" type="xs:float" />
<xs:attribute name="MaxLength" type="xs:int" />
<xs:attribute name="maxLength" type="xs:string" />
```


Copyright © 2014 Microsoft Corporation.

```
<xs:attribute name="Min" type="xs:string" />
<xs:attribute name="Mult" type="xs:string" />
<xs:attribute name="Name" type="xs:string" use="required" />
<xs:attribute name="NegativeFormat" type="xs:string" />
<xs:attribute name="node" type="xs:string" />
<xs:attribute name="Node" type="xs:string" />
<xs:attribute name="NoEditFormBreak" type="TRUEFALSE" default="FALSE" />
<xs:attribute name="NumLines" type="xs:string" default="6" />
<xs:attribute name="Percentage" type="TRUEFALSE" default="FALSE" />
<xs:attribute name="PIAttribute" type="xs:string" />
<xs:attribute name="PIN" type="TRUEFALSE" default="FALSE" />
<xs:attribute name="PITarget" type="xs:string" />
<xs:attribute name="PrependId" type="TRUEFALSE" default="FALSE" />
<xs:attribute name="Presence" type="TRUEFALSE" default="FALSE" />
<xs:attribute name="PreviousName" type="xs:string" />
<xs:attribute name="PrimaryKey" type="TRUEFALSE" />
<xs:attribute name="PrimaryPIAttribute" type="xs:string" />
<xs:attribute name="PrimaryPITarget" type="xs:string" />
<xs:attribute name="ReadOnly" type="TRUEFALSE" default="FALSE" />
<xs:attribute name="ReadOnlyEnforced" type="TRUEFALSE" default="FALSE" />
<xs:attribute name="RenderXMLUsingPattern" type="TRUEFALSE" default="FALSE" />
<xs:attribute name="Required" type="TRUEFALSE" default="FALSE" />
<xs:attribute name="RestrictedMode" type="TRUEFALSE" default="TRUE" />
<xs:attribute name="ResultType" type="FieldInternalType" />
<xs:attribute name="ResyncOnChange" type="TRUEFALSE" default="FALSE"/>
<xs:attribute name="RichText" type="TRUEFALSE" default="FALSE" />
<xs:attribute name="RichTextMode" type="FieldRichTextMode" default="Compatible" />
<xs:attribute name="RowOrdinal" type="xs:int" default="0" />
<xs:attribute name="RowOrdinal2" type="xs:int" default="0" />
<xs:attribute name="Sealed" type="TRUEFALSE" default="FALSE" />
<xs:attribute name="SeparateLine" type="TRUEFALSE" default="FALSE" />
<xs:attribute name="SetAs" type="xs:string" />
<xs:attribute name="ShowAddressBookButton" type="TRUEFALSE" default="FALSE" />
<xs:attribute name="ShowField" type="xs:string" />
<xs:attribute name="ShowInDisplayForm" type="TRUEFALSE" />
<xs:attribute name="ShowInEditForm" type="TRUEFALSE" default="TRUE" />
<xs:attribute name="ShowInFileDialog" type="TRUEFALSE" />
<xs:attribute name="ShowInFileDlg" type="TRUEFALSE" />
<xs:attribute name="ShowInListSettings" type="TRUEFALSE" />
<xs:attribute name="ShowInNewForm" type="TRUEFALSE" default="TRUE" />
<xs:attribute name="ShowInVersionHistory" type="TRUEFALSE" />
<xs:attribute name="ShowInViewForms" type="TRUEFALSE" />
<xs:attribute name="Sortable" type="TRUEFALSE" />
<xs:attribute name="SourceID" type="xs:string" />
<xs:attribute name="StaticName" type="xs:string" />
<xs:attribute name="StorageTZ" type="xs:string" />
<xs:attribute name="StripWS" type="xs:string" />
<xs:attribute name="SuppressNameDisplay" type="TRUEFALSE" />
<xs:attribute name="TextOnly" type="TRUEFALSE" />
<xs:attribute name="Title" type="xs:string" />
<xs:attribute name="TitleField" type="TRUEFALSE" default="FALSE" />
<xs:attribute name="Type" type="xs:string" use="required" />
<xs:attribute name="UniqueId" type="xs:string" />
<xs:attribute name="UnlimitedLengthInDocumentLibrary" type="TRUEFALSE" />
<xs:attribute name="URLEncode" type="TRUEFALSE" />
<xs:attribute name="URLEncodeAsURL" type="TRUEFALSE" />
<xs:attribute name="UserSelectionMode" type="xs:string" />
<xs:attribute name="UserSelectionScope" type="xs:int" />
<xs:attribute name="Version" type="xs:int" default="0" />
```


Copyright © 2014 Microsoft Corporation.

```
    <xs:attribute name="Viewable" type="FALSE_Case_Insensitive_Else_Anything" />
    <xs:attribute name="WebId" type="UniqueIdentifierWithoutBraces" />
    <xs:attribute name="Width" type="xs:int" />
    <xs:attribute name="WikiLinking" type="TRUEFALSE" default="FALSE" />
    <xs:attribute name="WorkflowStatusURL" type="xs:string" use="optional" />
    <xs:attribute name="XName" type="xs:string" />
    <xs:anyAttribute namespace="##other" processContents="lax" />
</xs:complexType>
```

2.2.8.3.3.2 Attributes

Aggregation: For fields with the Node or node attribute, a reader MUST use this attribute to control promotion from XML or site template files. For other fields, a reader MUST ignore this attribute.

aggregation: For fields whose Type attribute maps to the Note field (2) internal type, the reader MAY permit a filtering user interface if this attribute contains the value merge (with case-insensitive comparison<4>). Otherwise a reader MUST ignore this attribute. A writer SHOULD NOT include this attribute.

AllowDeletion: If this attribute is FALSE, then the server MUST NOT permit the field (2) to be removed from the schema of a list (1).

AllowHyperlink: For fields whose Type maps to the Note field (2) internal type, and for which RichText is TRUE and RichTextMode is Compatible, a reader can use this attribute to determine if the editing UI allows insertion of hyperlinks. If this attribute is TRUE, a server or client presenting an editing user interface for this field (2) MUST include the insert hyperlink command. Otherwise, the editing user interface MUST NOT include the insert hyperlink command.

AllowMultiVote: The reader MUST ignore this attribute.

AppendOnly: If TRUE, for fields whose Type maps to the Note field (2) internal type, a client or server which presents a way of editing the value for this field (2) MUST do so in a way that ensures preservation of old data in the field (2).

AuthoringInfo: Text describing the field (2) to schema authors. A client or server presenting a schema-editing UI can display the text in this attribute.

BaseRenderingType: Field internal type that the server uses to render a field (2). If not present, the Type attribute does not map to an internal type in a way suitable for rendering. The server MUST derive the value of this attribute from the Type attribute, overriding the client-specified value.

BaseType: Allows a schema author to override the internal storage format for choice fields. A reader MUST ignore this attribute unless the field's internal type is Choice, MultiChoice, or GridChoice. A writer MUST NOT set this attribute's value to Choice, MultiChoice, or GridChoice.

CalType: Associates a calendar type with a field (2). A reader SHOULD take this value into consideration when rendering the field or performing date calculations on it. If this attribute is not present, a reader MUST use the CalendarType attribute. If that is also not present, it SHOULD fall back to the user's preference, and if that is not specified, to the calendar type for the site (2).

CalendarType: This is equivalent to CalType. A reader MUST use the value of CalType if present, but if that attribute is not present, it MUST use this attribute for the same purpose. A writer SHOULD omit this attribute, using CalType instead where it is needed.

Calculated: The reader MUST ignore this attribute.

Copyright © 2014 Microsoft Corporation.

CanToggleHidden: If TRUE, a server MUST permit the user to change the value of the Hidden attribute. Otherwise, a server MUST prevent the Hidden attribute from changing.

CountRelated: If TRUE, the internal type of the field (2) MUST be Lookup, the List and FieldRef attributes of the field MUST refer to a field of internal type Lookup, and for the referred field, its List attribute MUST refer to the list of which the source field is a member. A writer MUST NOT include this attribute if the Mult attribute is present. A reader MUST ignore the attribute if the field (2) internal type of the field is not Lookup.

ClassInfo: A writer SHOULD use values Menu or Icon, or it SHOULD omit the attribute. When rendering the value of the field (2) in a view, a server SHOULD use this to format the field appropriately for a menu or an icon. The server MUST ignore values other than Menu or Icon and MUST fall back to the default behavior for other values.

ColName: A server SHOULD use this attribute to describe the physical storage location for the field (2). A client MUST ignore this attribute and a server MUST pick its own value for this attribute when accepting CAML from a client.

ColName2**:** This has the same restrictions and semantics as the ColName attribute except it describes a **secondary** physical storage location.

Commas: A writer SHOULD NOT use this attribute; a reader MUST ignore this attribute.

Customization: A writer SHOULD NOT use this attribute; a reader MUST ignore this attribute.

Decimals: The number of digits to render in the fractional part of a number. A writer SHOULD only use this attribute on fields whose field (2) internal type is Number or Calculated. The reader MUST ignore this attribute on files with other internal types. A writer SHOULD omit this attribute rather than write the default value of "-1". When rendering a numeric field (2), a server MUST use this attribute to determine the number of decimal places to render. For the special value of -1, trailing zeros MUST be omitted.

DefaultURLDesc: For a field (2) whose field internal type is URL, this is the value to use for the description part when the user does not provide a description. The default is to use the URL part of the field (2). If the field's internal type is not URL, a reader MUST ignore this attribute.

Description: Textual description of the field (2) to be displayed in the user interface for schema authoring.

Direction: Specifies that HTML field (2) rendering will be done in left-to-right, right-to-left, or in the ambient context of the surrounding page.

Dir: A writer SHOULD NOT use this attribute; a reader MUST ignore this attribute.

DisplaceOnUpgrade: When TRUE, indicates that a server depends on the name of a field (2). The client MUST NOT set this attribute.

DisplayImage: When specified, this attribute supplies the URL, relative to the server's /_layouts/images folder, to render when displaying the field (2). A reader MUST ignore this attribute except for CrossProjectLink fields and Recurrence fields. If not specified, a blank placeholder image is used instead.

DisplayName: The text to display in the user interface when referring to the field (2).

DisplayNameSrcField: When present, specifies the name of another field (2) on the list (1) to which the **DisplayName** of the current field is synchronized.

Copyright © 2014 Microsoft Corporation.

DisplaySize: Number of columns to offer when displaying the field (2).

Div: In fields whose field (2) internal type is Integer, Number, or Calculated, a reader MUST interpret this as a floating point number by which the value is divided at render time. In fields of other internal types, a reader MUST ignore this attribute and a writer MUST NOT include this attribute.

EnableLookup: Specifies whether the user interface for creating a Lookup field (2) lists a Computed field as a possible target for the lookup. Unless the target field (2) type is Computed, a writer SHOULD NOT include this attribute and a reader MUST ignore it.

ExceptionImage: Specifies an image to display in place of DisplayImage for recurrence fields for items that are exceptions to the recurrence. A writer MUST include this attribute for recurrence fields if the DisplayImage is also included and MUST NOT include it otherwise.

Ext: A writer SHOULD only include this attribute when the Format attribute is set to Telephone. A reader MUST ignore this unless the Format is set to Telephone.

FieldRef: When present on a Lookup field (2), specifies the Name of another field on the list (1) from which to obtain the local value for the lookup. Lookup fields for which this attribute is not specified will supply their own storage for the local value.

FillInChoice: Specifies whether or not a form (2) generated to let the user edit a choice field (2) or multichoice field allows values other than those listed in the CHOICES child element. A reader MUST ignore this attribute except for the Choice fields and MultiChoice fields.

Filterable: Specifies whether items can be omitted or included from the results of a query based on the value of the field (2). TRUE if the field can be filtered; FALSE otherwise.

FilterableNoRecurrence: When TRUE, and the Filterable attribute is FALSE, specifies that items can be included or omitted from the results of a query in views that do not expand recurring events.

ForcedDisplay: If present, specifies a value for the field (2) to display in place of the field's real value.

Format: Specifies how to render the field (2). The interpretation of this attribute and the range of legal values depend on the value of the Type attribute, as follows:

- For a **datetime** field (2), the value MUST be DateTime, DateOnly, TimeOnly, ISO8601, ISO8601Basic, ISO8601Gregorian, or not present. A reader MUST interpret a missing attribute to mean DateTime.
- For a **Choice** field (2), the value MUST be DropDown, RadioButtons, Checkbox, or not present. A reader MUST interpret a missing attribute to mean DropDown.
- For a **CrossProjectLink** field (2), the value MUST be EventList or not present.
- For URL fields, the value MUST be Image, Hyperlink, or not present. A reader MUST interpret a missing attribute as Hyperlink.
- For a **Text** field (2), the value MUST be Telephone or not present. The value Telephone indicates that the edit form of the field is to be rendered in a way to facilitate entry of telephone numbers. The absence of the attribute indicates no special editing mode.
- For a **Boolean** field (2), the value MUST be CheckboxIcons, CheckboxIconsWithHeaderIcon, or not present.
- A reader MUST ignore this attribute for other field (2) types.

Copyright © 2014 Microsoft Corporation.

FromBaseType: Indicates that the field (2) was inherited from the list's base type. If TRUE, a server prevents the field from being deleted or having its type changed.

GridEndNum: Specifies the largest choice possible in a GridChoice field (2). A writer MUST include this attribute for **GridChoice** fields. A reader MUST ignore this attribute for other field types.

GridNATxt: A textual value to render for the choice that indicates "not applicable" in a **GridChoice** field (2). A reader MUST ignore this attribute for fields of other types.

GridStartNum: Specifies the smallest choice possible in a **GridChoice** field (2). A writer MUST include this attribute for GridChoice fields. A reader MUST ignore this attribute for fields of other types. A writer SHOULD set the value of this attribute to 1.

GridTxtRng1: A textual value to render for the choice that indicates the value corresponding to that which is specified by the GridStartNum attribute in a **GridChoice** field (2). A reader MUST ignore this attribute for other field types.

GridTxtRng2: A textual value to render for the choice that indicates the middle value in a **GridChoice** field (2). A reader MUST ignore this attribute for other field types.

GridTxtRng3: A textual value to render for the choice that indicates the value corresponding to that specified by the GridEndNum attribute in a **GridChoice** field (2). A reader MUST ignore this attribute for fields of other types.

Group: Name of the field (2) that is used for grouping purposes. A reader MUST take a localized version of "Custom Columns" as the default value for this attribute.

HeaderImage: Specifies a URL, relative to the /_layouts/images folder on the server, to render in place of the field's DisplayName in views and forms (2).

Height: Height in pixels to render an image for a URL field (2) whose Format attribute is equal to Image. If not present, specifies that the target image does not need to be scaled.

A reader MUST ignore this attribute, except for URL fields where the Format attribute specifies Image and the Width attribute is present.

Hidden: Specifies whether or not to render a field (2) in views or forms (2).

ID: GUID for the field (2).

Id: A writer SHOULD NOT include this attribute. A reader MUST ignore this attribute.

IMEMode: If specified, indicates a value for the ime-mode attribute to be applied to an <input> tag when reading the field (2).

Indexed: Specifies that a server can optimize for queries that filter on this field (2).

ImnHeader: This attribute is a marker. A reader MUST only check for the existence of the attribute. A writer MUST either not include the attribute or set its value to TRUE.

IsolateStyles: For a Text field (2) whose RichText attribute is TRUE and whose RichTextMode attribute is FullHtml, this attribute specifies that a server will rewrite the HTML of the field to ensure it will not interfere with the rendering of the surrounding page. A reader MUST ignore this attribute in other circumstances.

JoinColName: Specifies the physical storage location in the list (1) referred to by the List attribute to compare with the local value of a Lookup field (2).

Copyright © 2014 Microsoft Corporation.

JoinRowOrdinal: A writer SHOULD NOT include this attribute<5>.

JoinType: Specifies how to treat items that have no corresponding matching item in the list (1) indicated by the List attribute. A reader MUST ignore this attribute except for Lookup fields.

JumpTo: Specifies the Name of the next field (2) in a survey list (1) when the field has a specified value. A reader MUST ignore this attribute for Boolean fields.

JumpToFillinChoice: Specifies the Name of the next field (2) in a survey list (1) when a value other than those listed in the CHOICES child element is selected. A reader MUST ignore this attribute except for Choice fields and **MultiChoice** fields for which the **FillInChoice** attribute is TRUE.

JumpToNo: Specifies the Name of the next field (2) in a survey list (1) for a Boolean field whose value is FALSE.

JumpToYes: Specifies the Name of the next field (2) in a survey list (1) for a Boolean field whose value is TRUE.

LCID: The LCID used to render Number fields, Currency fields, and DateTime fields. When not present, specifies that the default LCID for the user is to be used. A reader MUST ignore this attribute for other types of fields.

List: Specifies the foreign list (1) for a Lookup field (2). A writer MUST include this attribute for Lookup fields. A reader MUST ignore this attribute for other kinds of fields.

Max: Specifies the maximum value for a Number field (2). A reader MUST ignore this attribute for other kinds of fields.

MaxLength: Specifies the maximum number of characters allowed in a Text field (2).

maxLength: The reader MUST ignore this attribute.

Min: Specifies the minimum value for a Number field (2). A reader MUST ignore this attribute for other kinds of fields.

Mult: In fields whose field (2) internal type is Integer, Number, or Calculated, a reader MUST interpret this as a floating point number by which the value is multiplied at render time. In this context, a reader MUST interpret the default to be 1.0.

In the context of a Lookup field (2), a reader MUST interpret the presence of this attribute as an indication that the field is a multi-value lookup, and the absence of which is an indication that the field is a single-value lookup. In this context, a writer MUST either omit the attribute or set the value to TRUE. In other contexts, a reader MUST ignore the attribute and the writer SHOULD NOT include the attribute.

Name: String that identifies the field (2) within its list (1).

NegativeFormat: A writer SHOULD NOT include this attribute <6>. A reader MUST ignore this attribute.

node: A writer SHOULD use the Node attribute rather than this attribute. A reader MUST ignore this attribute when the Node attribute is present. Otherwise, a reader MUST use this attribute for the same purpose as it would use the Node attribute.

Node: When present, specifies an XPath to be used to read or write the value of the field (2) into an XML document.

Copyright © 2014 Microsoft Corporation.

NoEditFormBreak: For fields where RichText is TRUE or where the Type is Choice or MultiChoice, a reader MUST ignore this attribute. For other fields, if TRUE, this attribute specifies that the following field (2) is to be rendered on the same line as this field.

NumLines: The number of lines to render when accepting input for a Note field (2). A reader MUST ignore this attribute for other fields.

Percentage: When specified on a Number field (2) or a Calculated field that evaluates to a number, specifies that the field is rendered as a percentage (100 times its value and with a trailing "%" character). A reader MUST ignore this attribute for other fields.

PIAttribute: A reader MUST ignore this attribute if either the Node or node attribute is present, or if both the **PrimaryPIAttribute** and **PrimaryPITarget** attributes are present, or if the **PITarget** attribute is not present. Otherwise, the reader MUST treat this attribute as it would the **PrimaryPIAttribute** attribute.

PIN: A writer SHOULD only include this when the Format attribute is set to Telephone. A reader MUST ignore this unless the Format is set to Telephone.

PITarget: A reader MUST ignore this attribute if either the Node or node attribute is present, if both the **PrimaryPIAttribute** and **PrimaryPITarget** attributes are present, or if the **PIAttribute** attribute is not present.

Otherwise, the reader MUST treat this attribute as it would the **PrimaryPITarget** attribute.

PrependId: Specifies whether the edit form for a Lookup field (2) for which the Multi attribute is TRUE will present choices sorted by ID rather than by the ShowField. A reader MUST ignore this attribute in other circumstances.

Presence: Specifies whether a User field (2) will be decorated with instant messaging presence information. A reader MUST ignore this attribute for fields that are not user fields.

PreviousName: Indicates the value of the Name attribute of the field (2) in an earlier revision of the containing list's schema.

PrimaryKey: A reader MUST ignore this attribute.

PrimaryPIAttribute: When present, specifies an attribute for an XML processing instruction to be used to read or write the value of the field (2) into an XML document.

PrimaryPITarget: When present, specifies an XML processing instruction to be used to read or write the value of the field (2) into an XML document.

ReadOnly: Determines whether or not the user is allowed to change the field (2) through the user interface. If TRUE, only programmatic changes are allowed.

ReadOnlyEnforced: Determines whether or not the user is allowed to change the field (2) by any means. If TRUE, the field (2) can only be changed by the system.

RenderXMLUsingPattern: Specifies whether or not a Computed field (2) is rendered using its DisplayPattern child element even when rendering for the object model or SOAP interfaces. A reader MUST ignore this attribute except for Computed fields.

Required: Specifies whether or not forms (2) presented to accept data for the list item permits blank values for the field (2).

Copyright © 2014 Microsoft Corporation.

RestrictedMode: For Notes fields for which the RichText attribute is TRUE, this attribute specifies whether or not the field (2) permits cut, copy, paste, and insert image commands. For other kinds of fields, the reader MUST ignore this attribute.

ResultType: If the Type attribute specifies that the internal type of the field (2) is Calculated, a reader SHOULD use this attribute as in place of the internal type of the field to render results. A reader MUST ignore this attribute if the internal type of the field (2) is not Calculated.

ResyncOnChange: Specifies whether or not the field's value changes value on form (2) submission.

RichText: Specifies whether or not a Note field (2) contains formatted text; the exact text formatting is subject to the value of the RichTextMode attribute. A reader MUST ignore this attribute except for Note fields.

RichTextMode: Specifies the serialization of formatted text.

RowOrdinal: The comments for the **ColName** attribute apply to this attribute as well.

RowOrdinal2: The comments for the **ColName** attribute apply to this attribute as well.

Sealed: Specifies how the server allows the field (2) to be changed. If TRUE, the server MUST prevent changes to the field (2) except for the DisplayName, Description, Hidden, and Indexed attributes and the **ParserRefs** child element.

SeparateLine: Determines whether the field (2) is displayed on a different row for views that support this feature.

SetAs: A writer SHOULD NOT include this attribute. A reader MUST ignore this attribute.

ShowAddressBookButton: Determines whether or not a field's edit control includes a facility for entering users from an address book. A reader MUST ignore this attribute except when rendering the field (2) in the context of a form (2) that supports this functionality.

ShowField: Specifies the field (2) in the foreign list (1) that supplies the value of a Lookup field. A reader MUST ignore this attribute unless the field (2) is a Lookup field. If the Type attribute is User, the default value for this attribute is InmName. If the Type is WorkflowStatus, the default value for this attribute is Status1. Otherwise, the default value is Title.

ShowInDisplayForm: Whether the field (2) is shown or hidden in a form (2) designed to show the item in a read-only fashion. If TRUE, the field (2) will be shown in the form (2) for the item. If FALSE, the field (2) will not be shown in the form (2). If not specified, an implementation-specific algorithm is used to determine if the field will be shown.

ShowInEditForm: When FALSE, indicates that the field (2) is not included in the form (2) that is used to modify an item. When TRUE, indicates that the field's inclusion in such a form (2) depends on implementation.

ShowInFileDialog: The reader MUST ignore this attribute.

ShowInFileDlg: Similar to **ShowInEditForm**, except that it applies to a form (2) designed to collect information about a document from within the context of an application.

ShowInEditDlg: Similar to **ShowInEditForm**, except that it applies to a form (2) designed to collect information about an item that is being modified.

ShowInListSettings: Similar to **ShowInEditForm**, except that it applies to the field's inclusion in the user interface that is presented to list schema editors.

Copyright © 2014 Microsoft Corporation.

ShowInNewForm: Similar to **ShowInEditForm**, except that it applies to the field's inclusion in a form (2) designed to collect information about an item that is being created.

ShowInVersionHistory: Similar to **ShowInEditForm**, except that it applies to the field's inclusion in a form (2) designed to display a read-only rendition of a historical version of an item.

ShowInViewForms: Similar to **ShowInEditForm**, except that it applies to the field's inclusion in the user interface that is presented to view authors.

Sortable: When FALSE, specifies that query results are not allowed to be ordered with respect to this field (2).

SourceID: URI suitable for use in an **XML namespace** in cases where the list schema is transformed to an XSD.

StaticName: Local part of an XML element name that is unique within the namespace given by the **SourceID** attribute.

StorageTZ: For DateTime fields, specifies the time zone in which the field (2) is stored. If TRUE, UTC is indicated; otherwise the site's local time zone is indicated. The reader MUST ignore this attribute unless the field's type is DateTime.

StripWS: The reader MUST ignore this attribute.

SuppressNameDisplay: When specified on a User field (2), the user's name is not displayed but all other rendering information is displayed.

TextOnly: The reader MUST ignore this attribute.

Title: When present on a **CrossProjectLink** field (2) whose DisplayImage attribute is set and whose Format attribute specifies EventList, this attribute specifies a textual alternative to the image. When not present but the other conditions are specified, the reader MUST infer a default value from the DisplayName attribute. When these conditions are not met, the reader MUST ignore the attribute.

TitleField: Specifies that the file is suitable to use for a title of the item.

Type: Specifies the rendering properties and internal type of the field (2).

UniqueId: The reader MUST ignore this attribute.

UnlimitedLengthInDocumentLibrary: On a Note field (2) or on a Lookup field for which the Mult attribute is TRUE, this attribute specifies that the length of the content is not limited.

URLEncode: The reader MUST ignore this attribute.

URLEncodeAsURL: The reader MUST ignore this attribute.

UserSelectionMode: When present on a User field (2), PeopleOnly indicates that only principals that are users are selected, and PeopleAndGroups indicates that both principals that are users and those that are groups can be selected.

UserSelectionScope: When present on a User field (2), specifies the group to which the selected user or users belong.

Version: Current version of the field (2). The server MUST increment the value by 1 each time the field definition is changed. The client MUST ignore this attribute.

Copyright © 2014 Microsoft Corporation.

Viewable: Specifies whether or not the field (2) was added to the default view of the list (1) when it was added to the schema of the list (1). The reader SHOULD ignore this attribute.

WebId: For Lookup fields, this attribute specifies the identifier of the site (2) that contains the list (1) referenced by the List attribute.

Width: This attribute has the same semantics as the Height attribute except that it refers to the width of the image rather than the height.

WikiLinking: When TRUE on a Note field (2) whose RichText attribute is TRUE, additional processing is done to facilitate entering hyperlinks within the content of the field. In other circumstances, the reader MUST ignore this attribute.

WorkflowStatusURL: The reader MUST ignore this attribute.

XName: String that is used to correlate the field (2) in an external schema.

2.2.8.3.3.3 Child Elements

Customization**:** Provides an arbitrary XML document to provide vendor extensibility.

CHOICES: A set of CHOICE string elements that represent a list of available choices for a field (2). The reader MUST ignore CHOICES if the Type attribute is not Choice or MultiChoice. The writer SHOULD omit CHOICES if the Type attribute is not Choice or MultiChoice.<7>

Default: Default value of instances of data for this field (2) in a list item. The reader MUST ignore this element when the DefaultFormula element is present and not empty.

DefaultFormula: Formula used to calculate the default value for this field (2) in a list item.

DisplayBidiPattern: Specifies the rendering of a Computed field (2) to be used for a site (2) with a locale identifier that specifies a bidirectional read order. A reader MUST ignore **DisplayBidiPattern** if the Type attribute is not Computed. The writer SHOULD omit DisplayBidiPattern if the Type attribute is not Computed. <8>

DisplayPattern: Specifies the rendering of a Computed field (2). A reader MUST ignore DisplayPattern if the Type attribute is not Computed. The writer SHOULD omit **DisplayPattern** if the Type attribute is not Computed <9>

FieldRefs: Specifies fields needed to render **DisplayPattern** or **DisplayBidiPattern**. **DisplayPattern** and **DisplayBidiPattern** MUST NOT use fields that are not listed in this element.

Formula: Formula used to calculate a value for a list item based on the value of other fields in the list (1). Reader MUST ignore Formula if the Type attribute is not Calculated. Writer SHOULD omit Formula if the Type attribute is not Calculated <10>.

MAPPINGS: A set of MAPPING string elements that represents a canonical, language-agnostic identifier for a corresponding CHOICE with the value specified by the MAPPING element. The reader MUST ignore MAPPINGS if the Type attribute is not Choice or MultiChoice. The writer SHOULD omit MAPPINGS if the Type attribute is not Choice or MultiChoice<11>.

ParserRefs: Specifies alternate names for the field (2) when accessed by the parsers listed in child elements of this element.

Copyright © 2014 Microsoft Corporation.

2.2.8.3.4 FieldDefinitionDatabase Type

The **FieldDefinitionDatabase** type provides an overall container for field (2) definitions in a Content Database.

2.2.8.3.4.1 Schema

```
<xs:complexType name="FieldDefinitionDatabase">
  <xs:sequence>
    <xs:choice>
      <xs:element name="FieldRef" type="FieldRefDefinitionTP" />
      <xs:element name="Field" type="FieldDefinitionTP" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```

2.2.8.3.4.2 Attributes

None.

2.2.8.3.4.3 Child Elements

FieldRef: Specifies a reference to an existing field (2) definition that is used in the current list (1).

Field: Specifies either a definition of a new field (2) to be used in this list (1), or a reference to an existing field that was deleted.

2.2.8.3.5 FieldDefinitionDatabaseWithVersion Type

The FieldDefinitionDatabaseWithVersion type provides an overall container for field (2) definitions in **tp_Fields**, as specified in the Fields Information Result Set in section 3.1.5.29.1.

2.2.8.3.5.1 Schema

```
  <xs:complexType name="FieldDefinitionDatabaseWithVersion" mixed="true">
    <xs:all>
      <xs:element name="tp_Fields" type="FieldDefinitionDatabase" minOccurs="1"
maxOccurs="1"/>
    </xs:all>
  </xs:complexType>
```

2.2.8.3.5.2 Attributes

None.

2.2.8.3.5.3 Child Elements

tp_Fields: Specifies a collection of field (2) definitions that are stored in a content database.

<Content>: The body text of this element MUST contain a string with the following pattern and MUST precede any child elements.

Copyright © 2014 Microsoft Corporation.

```
([0-9][0-9]*[.]) ([0-9][0-9]*[.]) ([0-9] [0-9]*[.]) ([0-9][0-9]*[.]) ([0-9][0-9]*[.]) ([0-
9][0-9]*)
```

The first two groups MUST correspond to the major version and minor version of the product. The second two groups MUST correspond to the revision version and build number of the product. The last two groups MUST correspond to the revision of the field (2) definitions on the list (1) and the list's list template, respectively.

2.2.8.3.6 FieldDefinitionTP Type

The **FieldDefinitionTP** type specifies a field (2) and its associated components. **FieldDefinitionTP** has the same structure as **FieldDefinition**. However, if only the ID attribute of **FieldDefinitionTP** is specified, the element specifies a field (2) definition that was deleted by a user on a front-end Web Server.

2.2.8.3.6.1 Schema

```
<xs:complexType name="FieldDefinitionTP">
  <xs:complexContent>
    <xs:extension base="FieldDefinition">
      <xs:attribute name="Type" type="xs:string" use="optional" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

2.2.8.3.6.2 Attributes

See **FieldDefinition**

2.2.8.3.6.3 Child Elements

See **FieldDefinition**.

2.2.8.3.7 FieldParserRef Type

The **FieldParserRef** type allows schema authors to override the Name attribute of a FieldDefinition type for a specific parser.

2.2.8.3.7.1 Schema

```
<xs:complexType name="FieldParserRef">
  <xs:attribute name="Name" type="xs:string" />
  <xs:attribute name="ProgId" type="xs:string" />
</xs:complexType>
```

2.2.8.3.7.2 Attributes

Name: Specifies an alternative Name to provide to the parser specified by the **ProgId** attribute when describing the field (2).

Copyright © 2014 Microsoft Corporation.

ProgId: Specifies the parser to which to provide the alternative **Name**.

2.2.8.3.7.3 Child Elements

2.2.8.3.8 FieldParserRefs Type

The **FieldParserRefs** type allows schema authors to override the **Name** attribute of a **FieldDefinition** type for a collection of parsers.

2.2.8.3.8.1 Schema

```
    <xs:complexType name="FieldParserRefs">
      <xs:sequence>
        <xs:element name="ParserRef" type="FieldParserRef" minOccurs="0" maxOccurs="unbounded"
/>
      </xs:sequence>
    </xs:complexType>
```

2.2.8.3.8.2 Attributes

None.

2.2.8.3.8.3 Child Elements

ParserRef: Specifies the overridden Name for a specific parser.

2.2.8.3.9 FieldRefDefinitionField Type

The **FieldRefDefinitionField** type specifies field (2) definitions that are referenced within another field definition.

2.2.8.3.9.1 Schema

```
    <xs:complexType name="FieldRefDefinitionField" mixed="true" >
      <xs:attribute name="Name" type="xs:string" use="required" />
      <xs:attribute name="ID" type="UniqueIdentifierWithOrWithoutBraces" use="optional" />
      <xs:attribute name="ShowField" type="xs:string" use="optional" />
      <xs:attribute name="RefType" type="FieldRefType" use="optional" />
      <xs:attribute name="CreateURL" type="xs:string" use="optional" />
      <xs:attribute name="Key" type="xs:string" use="optional" />
      <xs:attribute name="DisplayName" type="xs:string" use="optional" />
    </xs:complexType>
```

2.2.8.3.9.2 Attributes

See the attributes section of **FieldDefinition**.

Name: Specifies the Name attribute of the referenced field (2).

ID: Specifies the ID attribute of the referenced fields. When both ID and Name are specified, the reader MUST use ID first and fall back to Name if the ID does not match.

Copyright © 2014 Microsoft Corporation.

ShowField: Specifies an alternate value for the **ShowField** attribute on a lookup field (2) when rendering in the context of a computed field.

RefType: Describes the type of reference of the field (2) in an events list. This MUST be a **FieldRefType**. In other cases, this attribute MUST NOT be present.

CreateURL: The URL to create a Meeting Workspace site (2). If the RefType is LinkURL, this attribute MUST be present. Otherwise it MUST NOT be present.

Key: If the value of this attribute is set to "Primary", the server MUST give this field (2) priority in the ordering of the items.

DisplayName: The reader MUST ignore this attribute.

2.2.8.3.9.3 Child Elements

<content>: Describes the Meeting Workspace site (2) created by the URL in CreateURL. If the RefType attribute is present and has the value LinkURL, then the element MUST have content. In other cases, the reader MUST ignore any content.

2.2.8.3.10 FieldRefDefinitionTP Type

The **FieldRefDefinitionTP** type specifies a reference to a field definition. The attributes specified here override existing values in the field definition.

2.2.8.3.10.1 Schema

```
<xs:complexType name="FieldRefDefinitionTP">
  <xs:attribute name="Name" type="xs:string" use="required" />
  <xs:attribute name="ColName" type="xs:string" use="optional" />
  <xs:attribute name="ColName2" type="xs:string" use="optional" />
  <xs:attribute name="RowOrdinal" type="xs:int" default="0" use="optional" />
  <xs:attribute name="RowOrdinal2" type="xs:int" default="0" use="optional" />
  <xs:attribute name="ID" type="UniqueIdentifierWithOrWithoutBraces" use="optional" />
  <xs:attribute name="SourceID" type="xs:string" use="optional" />
  <xs:attribute name="StaticName" type="xs:string" use="optional" />
</xs:complexType>
```

2.2.8.3.10.2 Attributes

See **FieldDefinition**.

2.2.8.3.10.3 Child Elements

None.

2.2.8.3.11 MAPPINGDEFINITION Type

The **MAPPINGDEFINITION** type defines a canonical value for localizable CHOICE entries. Each **MAPPINGDEFINITION** MUST define in its contents a corresponding value from a choice.

2.2.8.3.11.1 Schema

```
<xs:complexType name="MAPPINGDEFINITION">
```


Copyright © 2014 Microsoft Corporation.

```
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="Value" type="xs:string" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
```

2.2.8.3.11.2 Attributes

Value: String which contains a canonical, non-localizable value for a CHOICE.

2.2.8.3.11.3 Child Elements

<content>: Contains a string that MUST specify exactly the string of a corresponding CHOICE.

2.2.8.3.12 MAPPINGDEFINITIONS Type

The **MAPPINGDEFINITIONS** type is a container for one or more MAPPINGS. MAPPINGS MUST NOT be defined for fields other than Choice or MultiChoice. There MUST be either no MAPPINGDEFINITION elements defined for a Choice field (2), or exactly one MAPPING element for each corresponding CHOICE element.

2.2.8.3.12.1 Schema

```
    <xs:complexType name="MAPPINGDEFINITIONS">
      <xs:sequence>
        <xs:element name="MAPPING" type="MAPPINGDEFINITION" minOccurs="0" maxOccurs="unbounded"
/>
      </xs:sequence>
    </xs:complexType>
```

2.2.8.3.12.2 Attributes

None.

2.2.8.3.12.3 Child Elements

MAPPING: A canonical value mapping for a CHOICE.

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.

Copyright © 2014 Microsoft Corporation.

2.2.8.7 Attribute Groups

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

Copyright © 2014 Microsoft Corporation.

3 Protocol Details

3.1 Back End Database Server Details

3.1.1 Abstract Data Model

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

The BEDS maintains the following sets of data for this protocol within both a Configuration Database and one or more content databases. Data within the appropriate databases is maintained until updated or removed.

Configuration Objects: A set of information about farm configuration. The Configuration Objects are stored in a Configuration Database.

Site Map: A set of information mapping URLs for site collections to Site Collection Identifiers (section 2.2.1.9), and the content databases that contain each site collection's data. The Site Map is stored in the Configuration Database. Site Map entries are identified by Site Collection Identifiers and also are represented by either Absolute URLs or store relative form URLs.

Versions: A set of information indicating the current version information for various components in the farm.

Site Collections: A set of information about all site collections in a content database. Site collection entries are identified by Site Collection Identifiers and are also represented by either Absolute URLs or Store Relative Form URLs.

Sites: A set of information about all sites in a content database. Site entries are identified by Site Identifiers (section 2.2.1.11) and are also represented by Store Relative Form URLs.

Lists: A set of information about all lists in a content database. List entries are identified by List Identifiers (section 2.2.1.5) and are also represented by Store Relative Form URLs.

List Items: A set of information about all list items in a content database. List item entries are identified by List Item Identifiers (section 2.2.1.6).

Documents: A set of information about all documents in a content database. Document entries are identified by Document Identifiers (section 2.2.1.2) and are also are represented by Store Relative Form URLs.

Users: A set of information about all users in a content database. User entries are identified by User Identifiers (section 2.2.1.13).

Site Groups: A set of information about all site groups in a content database. Site group entries are identified by Site Group Identifiers (section 2.2.1.10).

Roles: A set of information about all roles in a content database. Role entries are identified by Role Identifiers.

Copyright © 2014 Microsoft Corporation.

3.1.2 Timers

An execution timeout timer is set up on the BEDS to govern the execution time for any requests. The amount of time is governed by a time-out value configured on the BEDS for all connections.

3.1.3 Initialization

Authentication of the TDS connection to the BEDS MUST occur before this protocol can be used.

The data structures, stored procedures, and actual data are persisted by the BEDS within databases, so any operations to initialize the state of the database MUST occur before the BEDS can use this protocol. This protocol requires that the data for site collections, sites, lists, and document libraries already exist within the BEDS in a valid state.

3.1.4 Higher-Layer Triggered Events

None.

3.1.5 Message Processing Events and Sequencing Rules

The T-SQL syntax for each Stored Procedure and Result Set, and the variables they are composed of, is defined using the T-SQL language specified in [TSQL-Ref]. 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 equal 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, because the WFE can access any column with no defined name by ordinal position. Such names are designated in the text using braces in the form {name}. For interoperability, named columns in Result Sets are specified with what they SHOULD be named, and columns marked with braces SHOULD have no defined name. WFE implementations MUST NOT rely on any column name in a Result Set.

The logical sequence of returned values and result sets are indicated in each of the individual Stored Procedures defined in this section. The TDS protocol controls the actual order and structure of how the T-SQL language formatted information is transported over the wire.

All functions, result sets, and stored procedures are defined using T-SQL.

The following table lists all functions and procedures that are defined in this section.

Function or procedure	Section
fn_GetFullUrl	3.1.5.1
proc_AddBuildDependency	3.1.5.2
proc_AddDocument	3.1.5.3
proc_AddListItem	3.1.5.4
proc_ChangeLevelForDoc	3.1.5.5
proc_CheckoutDocument	3.1.5.6

Copyright © 2014 Microsoft Corporation.

Function or procedure	Section
proc_ClearLinks	3.1.5.7
proc_CreateDir	3.1.5.8
proc_DeleteAllDocumentVersions	3.1.5.9
proc_DeleteDocBuildDependencySet	3.1.5.10
proc_DeleteDocumentVersion	3.1.5.11
proc_DeleteUrl	3.1.5.12
proc_DirtyDependents	3.1.5.13
proc_EnumLists	3.1.5.14
proc_FetchDocForHttpGet	3.1.5.15
proc_FetchDocForRead	3.1.5.16
proc_FetchDocForUpdate	3.1.5.17
proc_FetchWelcomeNames	3.1.5.18
proc_GenerateNextId	3.1.5.19
proc_GetAllAttachmentsInfo	3.1.5.20
proc_GetAuditMask	3.1.5.21
proc_GetAuditMaskOutput	3.1.5.22
proc_GetBlobIdsDocs	3.1.5.23
proc_GetBlobIdsVersions	3.1.5.24
proc_GetContainingList	3.1.5.25
proc_GetDocsMetaInfo	3.1.5.26
proc_GetLinkInfoSingleDoc	3.1.5.27
proc_GetListCheckedOutFiles	3.1.5.28
proc_GetListFields	3.1.5.29
proc_GetListMetaDataAndEventReceivers	3.1.5.30
proc_getObject	3.1.5.31
proc_getObjectsByBaseClass	3.1.5.32
proc_getObjectsByClass	3.1.5.33
proc_GetSiteFlags	3.1.5.34
proc_getSiteMap	3.1.5.35
proc_getSiteMapById	3.1.5.36

Copyright © 2014 Microsoft Corporation.

Function or procedure	Section
proc_GetTpWebMetaDataAndListMetaData	3.1.5.37
proc_GetUniqueScopesInList	3.1.5.38
proc_GetVersion	3.1.5.39
proc_GetWebMetaInfo	3.1.5.40
proc_GetWebMetainfoByUrl	3.1.5.41
proc_ListDocumentVersions	3.1.5.42
proc_ListUrls	3.1.5.43
proc_RenameUrl	3.1.5.44
proc_SecAddPrincipalToRole	3.1.5.45
proc_SecAddRoleDef	3.1.5.46
proc_SecAddUser	3.1.5.47
proc_SecAddUserToSiteGroup	3.1.5.48
proc_SecAddWebMembership	3.1.5.49
proc_SecChangeToInheritedList	3.1.5.50
proc_SecChangeToInheritedWeb	3.1.5.51
proc_SecChangeToUniqueScope	3.1.5.52
proc_SecCheckDeletedAccounts	3.1.5.53
proc_SecCloneRoleDefinitions	3.1.5.54
proc_SecCreateSiteGroup	3.1.5.55
proc_SecDecCurrentUsersCount	3.1.5.56
proc_SecGetAccountStatus	3.1.5.57
proc_SecGetAclFromScope	3.1.5.58
proc_SecGetAllAclsForSite	3.1.5.59
proc_SecGetAllGroupsAndMembershipInfo	3.1.5.60
proc_SecGetCompleteWebRoleMemberList	3.1.5.61
proc_SecGetCurrentUsersCount	3.1.5.62
proc_SecGetDomainGroupMapData	3.1.5.63
proc_SecGetGroupById	3.1.5.64
proc_SecGetGroupOwner	3.1.5.65
proc_SecGetGroupSecurityScopes	3.1.5.66

Copyright © 2014 Microsoft Corporation.

Function or procedure	Section
proc_SecGetIndividualUrlSecurityCheckEventReceivers	3.1.5.67
proc_SecGetPrincipalByEmail	3.1.5.68
proc_SecGetPrincipalById	3.1.5.69
proc_SecGetPrincipalByLogin	3.1.5.70
proc_SecGetPrincipalByLogin20	3.1.5.71
proc_SecGetPrincipalDisplayInformation20	3.1.5.72
proc_SecGetRoleAssignments	3.1.5.73
proc_SecGetRoleBindingsForAllPrincipals	3.1.5.74
proc_SecGetRoleDefs	3.1.5.75
proc_SecGetRoleDefsForPrincipal	3.1.5.76
proc_SecGetSecurityInfo	3.1.5.77
proc_SecGetSiteAdmins	3.1.5.78
proc_SecGetSiteGroupById	3.1.5.79
proc_SecGetSiteGroupByTitle	3.1.5.80
proc_SecGetSiteGroupByTitle20	3.1.5.81
proc_SecGetUserAccountDirectoryPath	3.1.5.82
proc_SecGetUserPermissionOnGroup	3.1.5.83
proc_SecListAllSiteMembers	3.1.5.84
proc_SecListAllWebMembers	3.1.5.85
proc_SecListGroupsInRole	3.1.5.86
proc_SecListScopeGroups	3.1.5.87
proc_SecListScopeUsers	3.1.5.88
proc_SecListSiteGroupMembership	3.1.5.89
proc_SecListSiteGroups	3.1.5.90
proc_SecListSiteGroupsContainingUser	3.1.5.91
proc_SecListSiteGroupsWhichUserOwns	3.1.5.92
proc_SecListUsersInRole	3.1.5.93
proc_SecMigrateUser	3.1.5.94
proc_SecReCalculateWebFGP	3.1.5.95
proc_SecRefreshToken	3.1.5.96

Copyright © 2014 Microsoft Corporation.

Function or procedure	Section
proc_SecRemoveGroup	3.1.5.97
proc_SecRemovePrincipalFromScope	3.1.5.98
proc_SecRemoveRoleDef	3.1.5.99
proc_SecRemoveUserFromScopeByLogin	3.1.5.100
proc_SecRemoveUserFromSite	3.1.5.101
proc_SecRemoveUserFromSiteGroup	3.1.5.102
proc_SecRemoveUserFromSiteGroupByLogin	3.1.5.103
proc_SecResetItemPerm	3.1.5.104
proc_SecResetWebToDefaultRoleDefinition	3.1.5.105
proc_SecResolvePrincipal	3.1.5.106
proc_SecSetSiteGroupProperties	3.1.5.107
proc_SecSetUserAccountDirectoryPath	3.1.5.108
proc_SecSetWebRequestAccess	3.1.5.109
proc_SecUpdateAnonymousPermMask	3.1.5.110
proc_SecUpdateDomainGroupMapData	3.1.5.111
proc_SecUpdateRoleDef	3.1.5.112
proc_SecUpdateUser	3.1.5.113
proc_SecUpdateUserActiveStatus	3.1.5.114
proc_TakeOverCheckOut	3.1.5.115
proc_UncheckoutDocument	3.1.5.116
proc_UpdateDocBuildDependencySet	3.1.5.117
proc_UpdateDocument	3.1.5.118
proc_UpdateListItem	3.1.5.119
proc_UpdateListSettings	3.1.5.120
proc_UpdateUserInfoInTableFromRowUpdater	3.1.5.121
proc_UrlToWebUrl	3.1.5.122

3.1.5.1 fn_GetFullUrl

The **fn_GetFullUr**l function is invoked to construct a full URL from two component parts.

```
FUNCTION fn_GetFullUrl(
      @DirName                        nvarchar(256),
```


Copyright © 2014 Microsoft Corporation.

```
      @LeafName                     nvarchar(128)
)
RETURNS                             nvarchar(260)
```

@DirName: The directory name component of the full URL.

@LeafName: The leaf name component of the full URL.

Return Values**:** The full URL, which is formed from **@DirName** and **@LeafName** as follows.

If **@DirName** is empty, then **@LeafName** MUST be returned. If **@LeafName** is empty, then **@DirName** MUST be returned. If either **@DirName** or **@LeafName** is NULL, then NULL MUST be returned. Otherwise **fn_GetFullUrl** MUST return **@DirName** + '/' + **@LeafName**.

3.1.5.2 proc_AddBuildDependency

The **proc_AddBuildDependency** stored procedure is invoked to associate a build dependency with a specified document.

```
PROCEDURE proc_AddBuildDependency(
      @DocSiteId                    uniqueidentifier,
      @DocDirName                   nvarchar(256),
      @DocLeafName                  nvarchar(128),
      @TargetDirName                nvarchar(256),
      @TargetLeafName               nvarchar(128),
      @DirectDependency             bit
);
```

@DocSiteId: The Site Collection Identifier (section 2.2.1.9) for the site (2) collection that contains the document. This parameter MUST NOT be NULL.

@DocDirName: The directory name of the document in store-relative form. This parameter MUST NOT be NULL.

@DocLeafName: The leaf name of the document. This parameter MUST NOT be NULL.

@TargetDirName: The directory name of the item to declare as a dependency of the document.

@TargetLeafName: The leaf name of the item to declare as a dependency of the document. This value MUST NOT be NULL.

@DirectDependency: Set to 1 if the target is a direct dependency. Set to 0 if indirect, such as a dependency of a dependency. This value MUST NOT be NULL.

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

The **proc_AddBuildDependency** stored procedure MUST NOT return a result set.

3.1.5.3 proc_AddDocument

The proc_AddDocument stored procedure is invoked to add a document to the BEDS with the specified parameters.

Copyright © 2014 Microsoft Corporation.

```
PROCEDURE proc_AddDocument(
      @DocSiteId                    uniqueidentifier,
      @DocWebId                     uniqueidentifier,
      @UserId                       int,
      @AuthorId                     int,
      @DocDirName                   nvarchar(256),
      @DocLeafName                  nvarchar(128)     OUTPUT,
      @Level                        tinyint,
      @UIVersion                    int               = 512,
      @NewDocId                     uniqueidentifier,
      @DoclibId                     uniqueidentifier,
      @NewDoclibRowId               int,
      @DocContent                   image,
      @DocMetaInfo                  image,
      @DocSize                      int,
      @DocMetainfoSize              int,
      @EnableMinorVersions          bit,
      @DocDirty                     bit,
      @DocFlags                     int,
      @DocIncomingCreatedDTM        datetime,
      @DocIncomingDTM               datetime,
      @GetWebListForNormalization   bit,
      @PutFlags                     int,
      @CreateParentDir              bit,
      @UrlIsSuggestion              bit,
      @ThicketMainFile              bit,
      @CharSet                      int,
      @ProgId                       nvarchar(255),
      @AttachmentOp                 int,
      @VirusVendorID                int,
      @VirusStatus                  int,
      @VirusInfo                    nvarchar(255),
      @LockTimeout                  int,
      @Comment                      nvarchar(1023),
      @@DocDTM                      datetime          OUTPUT,
      @fNoQuotaOrLockCheck          bit,
      @ChunkSize                    int,
      @DocTextptr                   varbinary(16)     OUTPUT
);
```

@DocSiteId: The Site Collection Identifier for the site collection that will contain the document to be stored. This MUST NOT be NULL.

@DocWebId: The Site Identifier (section 2.2.1.11) for the site (2) that will contain the document to be stored. This MUST NOT be NULL.

@UserId: The User Identifier (section 2.2.1.13) of the current user making the request to the WFE. This value MUST refer to an existing User Identifier for the specified site collection.

@AuthorId: The User Identifier to use instead of **@UserId's** value in the owner fields of the document if the publishing level is set to draft or checked out, or the document has a short-term lock applied. This can be NULL.

@DocDirName: The directory name of the document to be stored. This MUST NOT be NULL.

@DocLeafName: The leaf name of the document to be stored. If **@UrlIsSuggestion** is set to "1", this name can be replaced with a unique name and returned in this output parameter as the actual document leaf name. This MUST NOT be NULL.

Copyright © 2014 Microsoft Corporation.

@Level: The Publishing Level Type (section 2.2.2.6) value for the document to be stored. This MUST be a valid value.

@UIVersion: The UI version number to associate with this document. This MUST NOT be NULL.

@NewDocId: The document identifier (section 2.2.1.2) of the document to be stored. This MUST NOT be NULL and MUST be unique for a new document, and it MUST be the same for an existing document adding a new publishing level.

@DoclibId: The List Identifier (section 2.2.1.5) of the list (1) or document library in which the document is to be stored. This MUST only be NULL when a ghosted document is being added.

@NewDoclibRowId: The document library row identifier for the document to be stored. This can be NULL.

@DocContent: The optional document stream of the document. When this is not NULL, the @DocFlags parameter MUST have the 0x00000100 bit set to successfully store the document.

@DocMetaInfo: The metadata information for the document to be stored. This can be NULL.

@DocSize: Final size in bytes of the document stream image to be stored. This MUST be NULL if **@DocContent** is NULL.

@DocMetainfoSize: Size in bytes of the document's metadata info. This can be NULL if **@DocMetaInfo** is NULL.

@EnableMinorVersions: Specifies whether minor versions are enabled on the document. If this parameter is set to "1", minor versions MUST be enabled; otherwise, minor versions MUST NOT be enabled.

@DocDirty: Specifies whether the document has dependencies such as links to other items. If this parameter is set to "1", the document has dependencies that MUST subsequently be updated.

@DocFlags: A Doc Flags (section 2.2.2.3) value that contains options for adding the document.

@DocIncomingCreatedDTM: A time stamp in UTC format that specifies when the document was created.

@DocIncomingDTM: A time stamp in UTC format that specifies the document's last modification date.

@GetWebListForNormalization: Specifies whether to return the Site List for Normalization Result Set. If this parameter is set to "1", **proc_AddDocument** MUST return a list of the subsites of the document's containing site (2) in the Site List for Normalization Result Set (section 3.1.5.3.1).

@PutFlags: A Put Flags Type (section 2.2.2.7) value that specifies the options for adding the document.

@CreateParentDir: Specifies whether to create the parent directory for the document to be added if it does not already exist. If this parameter is set to "1", the parent directory specified by @DocDirName MUST be created if it does not exist. If this parameter is set to "0", proc_AddDocument MUST fail if the parent directory does not exist.

@UrlIsSuggestion: Specifies whether the **@DocLeafName** provided can be changed to a unique value if it is not unique. If this parameter is set to "1", **@DocLeafName** MUST be updated to get a guaranteed unique URL. If this parameter is set to "0" and the URL is not unique, this procedure MUST fail.

Copyright © 2014 Microsoft Corporation.

@ThicketMainFile: Specifies whether the document is a **thicket main file**. If this parameter is set to "1", then the document is a thicket main file.

@CharSet: An optional parameter that specifies a windows code page identifier for the character set to be associated with the document.

@ProgId: An optional parameter that specifies a preferred application to open the document. The **@ProgId** value is used to distinguish between different applications that save files with a given file extension (for example, different editors for HTML or XML files).

@AttachmentOp: An Attachments Flag (section 2.2.3.1) value that specifies the security checks to be performed by **proc_AddDocument** on this Document's URL, based on whether it appears to be an attachment.

@VirusVendorID: An optional parameter that specifies the identifier of the virus scanner that processed this document.

@VirusStatus: A Virus Status (section 2.2.3.18) type that specifies the current virus check status of this document.

@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 the document does not exist or if the document has not been processed by a virus scanner.

@LockTimeout: An integer value that specifies the number of minutes to set remaining on a short-term lock on the document. This value MUST be a positive value if a short-term lock is requested, and **proc_AddDocument** MUST set the short-term lock for this period. Otherwise, **proc_AddDocument** MUST NOT set a short-term lock on the document.

@Comment: An optional text check in comment to associate with the document. This value MUST be ignored when **@Level** is set to "255."

@@DocDTM: An output parameter for the time stamp of the last modification date of the document. This parameter MUST be set to the value of **@DocIncomingDTM** or to the current UTC date if **@DocIncomingDTM** is NULL.

@fNoQuotaOrLockCheck: Specifies whether to bypass the disk quota and disk lock check. If this parameter is set to "1", the checks are bypassed. If this parameter is set to "0", an explicit check will be made to see if the site (2) is locked or if quota is reached.

@ChunkSize: Specifies the size in bytes of the portion of the document stream in **@DocContent**. If **@ChunkSize** is less than **@DocSize** and a document stream is submitted, **@DocTextptr** MUST be returned to complete filling of the document stream image later. This parameter MUST be NULL if **@DocContent** is NULL; otherwise, it MUST NOT be NULL.

@DocTextptr: An output parameter containing a pointer set to the storage location of **@DocContent** if execution is successful, **@DocContent** is not NULL, and **@DocSize** is greater than **@ChunkSize**.

Return Values**:** The **proc_AddDocument** stored procedure returns an integer return code, which MUST be one of the values listed in the following table.

Value	Description
0	Successful execution.
3	The path specified for the document was not found.

Copyright © 2014 Microsoft Corporation.

Value	Description
5	Access denied error while attempting to create a directory.
80	Attempted to create a directory when the specified directory exists. The document already exists. The attempt to add the document failed. A document stream was specified, but the document is not checked out. The attempt to add the document stream failed.
212	The disk storage was locked.
1816	Disk quota exceeded.

The **proc_AddDocument** stored procedure returns zero to two result sets in the order listed.

3.1.5.3.1 Site List for Normalization Result Set

The Site List for Normalization Result Set returns a list of URLs for the immediate child subsites of the site (2) containing the newly added document. The Site List for Normalization Result Set MUST be produced when the input parameter **@GetWebListForNormalization** is set to "1" and execution has been successful up to the point of inserting the document. The Site List for Normalization Result Set MUST contain one row for each subsite found. The Site List for Normalization Result Set is defined in the URL Result Set (section 2.2.5.26) section.

3.1.5.3.2 Checkout Information Result Set

The Checkout Information Result Set returns check-out information about the document. The Checkout Information Result Set MUST be returned on successful completion if either the input parameter **@Level** is set to "255", indicating that the document is checked out, or if the input parameter **@LockTimeout** is not NULL, indicating that a short-term lock was applied. The Checkout Information Result Set MUST contain one row.

```
tp_Login                    nvarchar(255),
CheckoutDate                datetime,
{CheckoutExpires}           datetime;
```

tp_Login: The login name of the principal (1) to whom the document is checked out. If the document is currently checked in, this MUST be NULL.

CheckoutDate: A time stamp, in UTC format, indicating when this document was checked out. If the document is currently checked in, this MUST be NULL.

{CheckoutExpires}: A time stamp, in UTC format, indicating when the short-term lock for this document will expire. If the document is currently checked in or has a long-term checkout, this MUST be NULL.

3.1.5.4 proc_AddListItem

The **proc_AddListItem** stored procedure is invoked to add a list item to a list (1) or document library.

```
PROCEDURE proc_AddListItem (
      @SiteId                       uniqueidentifier,
```


Copyright © 2014 Microsoft Corporation.

```
@WebId                          uniqueidentifier,
@ListID                         uniqueidentifier,
@RowOrdinal                     int,
@ItemId                         int                             OUTPUT,
@UserId                         int,
@Size                           int,
@TimeNow                        datetime,
@CopySecurityFromMasterID       int                 = NULL,
@ExtraItemSize                  int                 = NULL,
@CheckDiskQuota                 bit                 = 1,
@ItemDocType                    tinyint             = 0,
@BaseRowItemId                  int                 = NULL,
@DocIdAdded                     uniqueidentifier    = NULL,
@RetainId                       uniqueidentifier    = NULL,
@RetainObjectIdentity           bit                 = 0,
@Level                          tinyint             = 1,
@UIVersion                      int                 = 512,
@ItemCountDelta                 int                 = 1,
@ItemName                       nvarchar(255)       = NULL,
@UseNvarchar1ItemName           bit                 = 1,
@ItemDirName                    nvarchar(256)       = NULL      OUTPUT,
@ItemLeafName                   nvarchar(128)       = NULL      OUTPUT,
@ServerTemplate                 int                 = NULL,
@IsNotUserDisplayed             bit                 = NULL,
@BaseType                       int                 = NULL,
@CheckSchemaVersion             int                 = NULL,
@OnRestore                      bit                 = 0,
@AddNamespace                   bit                 = 0,
@tp_Ordering                    varchar(512)        = NULL,
@tp_ThreadIndex                 varbinary(512)      = NULL,
@tp_HasAttachment               bit                 = NULL,
@tp_ModerationStatus            int                 = 0,
@tp_IsCurrent                   bit                 = 1,
@tp_ItemOrder                   float               = NULL,
@tp_InstanceID                  int                 = NULL,
@tp_GUID                        uniqueidentifier    = NULL,
@tp_Id                          int                 = NULL,
@tp_Author                      int                 = NULL,
@tp_Editor                      int                 = NULL,
@tp_Modified                    datetime            = NULL,
@tp_Created                     datetime            = NULL,
@tp_Version                     int                 = 1,
@tp_ContentType                 nvarchar(255)       = NULL,
@tp_ContentTypeId               varbinary(512)      = NULL,
@tp_CopySource                  nvarchar(260)       = NULL,
@tp_HasCopyDestinations         bit                 = NULL,
@tp_WorkflowVersion             int                 = 1,
@tp_WorkflowInstanceID          uniqueidentifier    = NULL,
@nvarchar1                      nvarchar(255)       = NULL,
@nvarchar2                      nvarchar(255)       = NULL,
@nvarchar3                      nvarchar(255)       = NULL,
@nvarchar4                      nvarchar(255)       = NULL,
@nvarchar5                      nvarchar(255)       = NULL,
@nvarchar6                      nvarchar(255)       = NULL,
@nvarchar7                      nvarchar(255)       = NULL,
@nvarchar8                      nvarchar(255)       = NULL,
@nvarchar9                      nvarchar(255)       = NULL,
@nvarchar10                     nvarchar(255)       = NULL,
@nvarchar11                     nvarchar(255)       = NULL,
```


Copyright © 2014 Microsoft Corporation.

```
@nvarchar12                 nvarchar(255)      = NULL,
@nvarchar13                 nvarchar(255)      = NULL,
@nvarchar14                 nvarchar(255)      = NULL,
@nvarchar15                 nvarchar(255)      = NULL,
@nvarchar16                 nvarchar(255)      = NULL,
@nvarchar17                 nvarchar(255)      = NULL,
@nvarchar18                 nvarchar(255)      = NULL,
@nvarchar19                 nvarchar(255)      = NULL,
@nvarchar20                 nvarchar(255)      = NULL,
@nvarchar21                 nvarchar(255)      = NULL,
@nvarchar22                 nvarchar(255)      = NULL,
@nvarchar23                 nvarchar(255)      = NULL,
@nvarchar24                 nvarchar(255)      = NULL,
@nvarchar25                 nvarchar(255)      = NULL,
@nvarchar26                 nvarchar(255)      = NULL,
@nvarchar27                 nvarchar(255)      = NULL,
@nvarchar28                 nvarchar(255)      = NULL,
@nvarchar29                 nvarchar(255)      = NULL,
@nvarchar30                 nvarchar(255)      = NULL,
@nvarchar31                 nvarchar(255)      = NULL,
@nvarchar32                 nvarchar(255)      = NULL,
@nvarchar33                 nvarchar(255)      = NULL,
@nvarchar34                 nvarchar(255)      = NULL,
@nvarchar35                 nvarchar(255)      = NULL,
@nvarchar36                 nvarchar(255)      = NULL,
@nvarchar37                 nvarchar(255)      = NULL,
@nvarchar38                 nvarchar(255)      = NULL,
@nvarchar39                 nvarchar(255)      = NULL,
@nvarchar40                 nvarchar(255)      = NULL,
@nvarchar41                 nvarchar(255)      = NULL,
@nvarchar42                 nvarchar(255)      = NULL,
@nvarchar43                 nvarchar(255)      = NULL,
@nvarchar44                 nvarchar(255)      = NULL,
@nvarchar45                 nvarchar(255)      = NULL,
@nvarchar46                 nvarchar(255)      = NULL,
@nvarchar47                 nvarchar(255)      = NULL,
@nvarchar48                 nvarchar(255)      = NULL,
@nvarchar49                 nvarchar(255)      = NULL,
@nvarchar50                 nvarchar(255)      = NULL,
@nvarchar51                 nvarchar(255)      = NULL,
@nvarchar52                 nvarchar(255)      = NULL,
@nvarchar53                 nvarchar(255)      = NULL,
@nvarchar54                 nvarchar(255)      = NULL,
@nvarchar55                 nvarchar(255)      = NULL,
@nvarchar56                 nvarchar(255)      = NULL,
@nvarchar57                 nvarchar(255)      = NULL,
@nvarchar58                 nvarchar(255)      = NULL,
@nvarchar59                 nvarchar(255)      = NULL,
@nvarchar60                 nvarchar(255)      = NULL,
@nvarchar61                 nvarchar(255)      = NULL,
@nvarchar62                 nvarchar(255)      = NULL,
@nvarchar63                 nvarchar(255)      = NULL,
@nvarchar64                 nvarchar(255)      = NULL,
@int1                       int                = NULL,
@int2                       int                = NULL,
@int3                       int                = NULL,
@int4                       int                = NULL,
@int5                       int                = NULL,
@int6                       int                = NULL,
```


Copyright © 2014 Microsoft Corporation.

```
@int7                   int             = NULL,
@int8                   int             = NULL,
@int9                   int             = NULL,
@int10                  int             = NULL,
@int11                  int             = NULL,
@int12                  int             = NULL,
@int13                  int             = NULL,
@int14                  int             = NULL,
@int15                  int             = NULL,
@int16                  int             = NULL,
@float1                 float           = NULL,
@float2                 float           = NULL,
@float3                 float           = NULL,
@float4                 float           = NULL,
@float5                 float           = NULL,
@float6                 float           = NULL,
@float7                 float           = NULL,
@float8                 float           = NULL,
@float9                 float           = NULL,
@float10                float           = NULL,
@float11                float           = NULL,
@float12                float           = NULL,
@datetime1              datetime        = NULL,
@datetime2              datetime        = NULL,
@datetime3              datetime        = NULL,
@datetime4              datetime        = NULL,
@datetime5              datetime        = NULL,
@datetime6              datetime        = NULL,
@datetime7              datetime        = NULL,
@datetime8              datetime        = NULL,
@bit1                   bit             = NULL,
@bit2                   bit             = NULL,
@bit3                   bit             = NULL,
@bit4                   bit             = NULL,
@bit5                   bit             = NULL,
@bit6                   bit             = NULL,
@bit7                   bit             = NULL,
@bit8                   bit             = NULL,
@bit9                   bit             = NULL,
@bit10                  bit             = NULL,
@bit11                  bit             = NULL,
@bit12                  bit             = NULL,
@bit13                  bit             = NULL,
@bit14                  bit             = NULL,
@bit15                  bit             = NULL,
@bit16                  bit             = NULL,
@uniqueidentifier1      uniqueidentifier = NULL,
@ntext1                 ntext           = NULL,
@ntext2                 ntext           = NULL,
@ntext3                 ntext           = NULL,
@ntext4                 ntext           = NULL,
@ntext5                 ntext           = NULL,
@ntext6                 ntext           = NULL,
@ntext7                 ntext           = NULL,
@ntext8                 ntext           = NULL,
@ntext9                 ntext           = NULL,
@ntext10                ntext           = NULL,
@ntext11                ntext           = NULL,
@ntext12                ntext           = NULL,
```


Copyright © 2014 Microsoft Corporation.

```
        @ntext13                    ntext           = NULL,
        @ntext14                    ntext           = NULL,
        @ntext15                    ntext           = NULL,
        @ntext16                    ntext           = NULL,
        @ntext17                    ntext           = NULL,
        @ntext18                    ntext           = NULL,
        @ntext19                    ntext           = NULL,
        @ntext20                    ntext           = NULL,
        @ntext21                    ntext           = NULL,
        @ntext22                    ntext           = NULL,
        @ntext23                    ntext           = NULL,
        @ntext24                    ntext           = NULL,
        @ntext25                    ntext           = NULL,
        @ntext26                    ntext           = NULL,
        @ntext27                    ntext           = NULL,
        @ntext28                    ntext           = NULL,
        @ntext29                    ntext           = NULL,
        @ntext30                    ntext           = NULL,
        @ntext31                    ntext           = NULL,
        @ntext32                    ntext           = NULL,
        @sql_variant1               sql_variant     = NULL,
        @error_sql_variant1         int             = 0,
        @sql_variant2               sql_variant     = NULL,
        @error_sql_variant2         int             = 0,
        @sql_variant3               sql_variant     = NULL,
        @error_sql_variant3         int             = 0,
        @sql_variant4               sql_variant     = NULL,
        @error_sql_variant4         int             = 0,
        @sql_variant5               sql_variant     = NULL,
        @error_sql_variant5         int             = 0,
        @sql_variant6               sql_variant     = NULL,
        @error_sql_variant6         int             = 0,
        @sql_variant7               sql_variant     = NULL,
        @error_sql_variant7         int             = 0,
        @sql_variant8               sql_variant     = NULL,
        @error_sql_variant8         int             = 0,
        @eventData                  image           = NULL,
        @acl                        image           = NULL
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) for the site collection containing the list (1) that the list item is being added to.

@WebId: The Site Identifier (section 2.2.1.11) for the site (2) containing the list (1) that the list item is being added to.

@ListID: The List Identifier (section 2.2.1.5) of the list (1) that the list item is being added to.

@RowOrdinal: The 0-based ordinal index of the current row to add for this list item in the set of rows representing the list item in the AllUserData (section 2.2.7.3) table. If a list item requires multiple rows to represent it in the AllUserData table because it contains more defined data columns than will fit in a single row, the WFE MUST call **proc_AddListItem** again, with the additional data using the next row value in the **@RowOrdinal** parameter. This parameter MUST NOT be NULL.

@ItemId: An output parameter that returns the identifier of the list item that has been added**:**

- If **@tp_Id** is not NULL, **proc_AddListItem** MUST use the value specified by **@tp_Id**.

Copyright © 2014 Microsoft Corporation.

- If **@BaseRowItemId** is not NULL, **proc_AddListItem** MUST use the value specified by **@BaseRowItemId**.
- If **@tp_Id** and **@BaseRowItemId** are NULL and **@ItemId** is not NULL, proc_AddListItem MUST use the value specified by **@ItemId**.
- If **@tp_Id**, **@BaseRowItemId** and **@ItemId** are all NULL, **proc_AddListItem** MUST generate a new value for the List Item Identifier (section 2.2.1.6) that is unique within the AllUserData table.

@UserId: The User Identifier (section 2.2.1.13) for the current user. proc_AddListItem uses this for purposes of permission-checking. This value MUST refer to an existing User Identifier for the specified site collection.

@Size: The size in bytes of the list item row to be added. This parameter MUST NOT be NULL.

@TimeNow: The current time, in UTC format, on the BEDS.

@CopySecurityFromMasterID: Specifies the OPTIONAL identifier of the list item to copy the scope security settings from for this list item. A list item that represents an exception to a recurrence item in a Meetings List (1) (that is, a list with a List Server Template Type of 200) MUST have the same scope security settings as the master recurrence item. If this parameter is set to a master recurrence item's List Item Identifier, then if the list item to be added does not share the scope of the master recurrence item, the new list item MUST be set to have a unique scope with a copy of the security settings from the scope of the master recurrence item. This parameter MUST only be set for list items that are exceptions to recurrence items in a Meetings List (1).

@ExtraItemSize: The size of the predefined SQL parameter fields in the list item row being added.

@CheckDiskQuota: A bit flag specifying whether Disk Quota needs to be checked for this user before adding the list item.

@ItemDocType: The Document Store Type (section 2.2.2.4) of the list item being added to the list (1).

@BaseRowItemId: An OPTIONAL value specifying the identifier of the list item to be added, if a value is not supplied by the **@tp_**Id parameter. If the **@tp_**Id parameter is NULL and this parameter is not NULL, **proc_AddListItem** MUST use this value for the identifier of the list item to be added.

@DocIdAdded: The OPTIONAL document identifier (section 2.2.1.2) of the document to be inserted as a list item in the list (1), if one exists. If **@DocIdAdded** is not NULL and the document has a document identifier (2), this parameter MUST be the existing document identifier (2). If this parameter is NULL, a new document identifier (2) MUST be generated for the list item. If this list item is being restored to the list (1) as part of a backup restore operation, as specified by the value of **@RetainObjectIdentity**, this parameter MUST be ignored, and the value of the **@RetainId** parameter MUST be used as the document identifier (2).

@RetainId: The document identifier (2) of the document to be inserted as a list item in the list (1), if this list item is being restored to the list (1) as part of a back-up restore operation.

@RetainObjectIdentity: A bit flag specifying whether this list item is being restored to the list (1) as part of a back-up restore operation. If **@RetainObjectIdentity** is set to 1, this list item is being restored to the list (1) as part of a back-up restore operation and MUST have the **@DocIdAdded** value specified in **@RetainId**.

Copyright © 2014 Microsoft Corporation.

@Level: The Publishing Level Type (section 2.2.2.6) value specifying the publishing status of this list item.

@UIVersion: The UI version number for the list item.

@ItemCountDelta: The number to be added to the list item count of the containing list (1):

- For a list item added with a single call to **proc_AddListItem**, or for one call for a list item with multiple rows to be added, this value MUST be 1.
- For the other calls to **proc_AddListItem** for the additional rows to be added for the list item, this value MUST be 0.

@ItemName: The display name of the list item.

@UseNvarchar1ItemName: If **@ItemName** is NULL, this bit flag specifies whether to use the content of **@nvarchar1** for the display name of the list item.

@ItemDirName: An output parameter containing the directory name of the list item.

@ItemLeafName: An output parameter containing the leaf name of the list item.

@ServerTemplate: The identifier for the List Server Template defining the base structure of the list (1) containing this list item.

@IsNotUserDisplayed: A bit flag specifying whether the user name is not displayed with list items.

@BaseType: The List Base Type (section 2.2.3.11) of the list (1) containing the list item.

@CheckSchemaVersion: This specifies an OPTIONAL schema version number to compare with the list schema version number. If this parameter is not NULL, the version numbers MUST match for successful completion.

@OnRestore: A bit flag that specifies whether this list item is being inserted by a back-up restore operation.

@AddNamespace: A Boolean value specifying whether metadata is being added to the list item. This parameter MUST NOT be NULL.

@tp_Ordering: This parameter specifies the threading structure for this list item in a previous discussion board list (1) (a list with a List Base Type of 3) as a concatenation of time stamp values in yyyyMMddHHmmss format. For all list items in lists with other List Base Types, this parameter MUST be NULL.

@tp_ThreadIndex: This specifies the list item's position within a threaded discussion board list (1) (a list with a List Base Type of 3) as a binary structure. For all list items in lists with other List Base Types, this parameter MUST be NULL.

@tp_HasAttachment: A bit flag that specifies whether the list item has an associated attachment.

@tp_ModerationStatus: A Moderation Status (section 2.2.3.13) value specifying the current moderation approval status of this list item.

@tp_IsCurrent: A bit flag that specifies whether this is the current version of this publishing level of the list item.

Copyright © 2014 Microsoft Corporation.

@tp_ItemOrder: This specifies the relative positioning order in which to view the list item when displayed with other List Items from the same list (1). This value can be the same as that of other List Items in the list (1).

@tp_InstanceID: If this list item is associated with a particular instance of a recurring meeting, this specifies the integer ID of that instance. For all other list items, this parameter MUST be NULL.

@tp_GUID: A List Item Identifier value that uniquely identifies this list item within the AllUserData table.

@tp_Id: The OPTIONAL integer identifier specified for this list item. If this parameter is not NULL, **proc_AddListItem** MUST use this value for the identifier of the list item to be added.

@tp_Author: The User Identifier for the user who created the list item.

@tp_Editor: The User Identifier for the user who last edited the list item.

@tp_Modified: A time stamp, in UTC format, that specifies when this list item was last modified.

@tp_Created: A time stamp, in UTC format, that specifies when this list item was created.

@tp_Version: Specifies the value to set for the counter used for internal conflict detection.

@tp_ContentType: Specifies the name of the content type for this list item.

@tp_ContentTypeId: Specifies the ID of the content type for this list item.

@tp_CopySource: Specifies the URL used as a source for this list item. If this list item was not copied from a source list item, this value MUST be NULL.

@tp_HasCopyDestinations: A bit flag specifying whether destination locations have been set for this list item to be copied to. If this list item does not have a destination location set, this value MUST be false.

@tp_WorkflowVersion: If this list item is part of a workflow (2), this parameter specifies the value to set denoting the state of this list item within that workflow (2). If this list item is not part of a workflow (2), this MUST be NULL.

@tp_WorkflowInstanceID: A Workflow Identifier (section 2.2.1.16) that specifies the currently active workflow instance on this list item. If this item is not part of a workflow (2), this MUST be NULL.

The next nine columns are duplicated a variable number of times, depending on the List Item's content type within the view definition, with each column referring to a separate List Server Template-defined field (2) or user-defined field within the containing list (1). Each instance of these individual column names is differentiated by a suffix with a value indicated in the column description, which replaces the placeholder '#' symbol shown as follows.

@nvarchar#: User-defined columns in the list (1) containing values of type nvarchar. There are 64 columns numbered from 1 to 64. If the column contains no data, the value MUST be NULL.

@int#: User-defined columns in the list (1) containing values of type int. There are 16 columns numbered from 1 to 16. If the column contains no data, the value MUST be NULL.

@float#: User-defined columns in the list (1) containing values of type float. There are 12 columns numbered from 1 to 12. If the column contains no data, the value MUST be NULL.

Copyright © 2014 Microsoft Corporation.

@datetime#: User-defined columns in the list (1) containing values of type datetime. There are eight columns numbered from 1 to 8. If the column contains no data, the value MUST be NULL.

@bit#: User-defined columns in the list (1) containing values of type bit. There are 16 columns numbered from 1 to 16. If the column contains no data, the value MUST be NULL.

@uniqueidentifier1: A user-defined column in the list (1) containing values of type **uniqueidentifier**. If the column contains no data, the value MUST be NULL.

@ntext#: User-defined columns in the list (1) containing values of type **ntext**. There are 32 columns numbered from 1 to 32. If the column contains no data, the value MUST be NULL.

@sql_variant#: User-defined columns in the list (1) containing values of type **sql_variant**. There are eight columns numbered from 1 to 8. If the column contains no data, the value MUST be NULL.

@error_sql_variant#: An integer specifying the type to be applied to the corresponding values specified as arguments for the parameter **@sql_variant#**. The possible types are listed in the following table.

Value	Description
1	Convert the argument value to a varbinary(2).
2	Convert the argument value to a bit.
3	Convert the argument value to a float.
4	Convert the argument value to a datetime.

@eventData: Contains implementation-specific event data significant to the WFE, but otherwise opaque to the BEDS, to be stored by the BEDS for eventual writing to a log file.

@acl: The binary serialization of the WSS ACL Format (section 2.2.4.6) ACL for the data supplied in **@eventData**, to be stored with the data.

Return Values: The **proc_AddListItem** stored procedure returns an integer return code, which MUST be listed in the following table, or MUST be a SQL Server **@@Error** result value.

Value	Description
0	Successful execution.
2	Postprocessing of the list item failed because a prerequisite list item was not found.
3	The directory specified for the list item does not exist.
5	The attempt to create a directory or document failed because the user does not have sufficient permissions.
13	The list item to be added is not valid.
16	Adding the list item caused updating of an existing list item to fail.
33	Cannot move directories that contain checked-out files.
50	A list item could not be deleted.
80	The document being added to the list already exists.

Copyright © 2014 Microsoft Corporation.

Value	Description
87	Unable to add the list item because the input parameters do not match existing list items, or an error occurred during a table update operation.
138	The list (1) could not be moved to the specified location.
160	Could not create a unique filename.
161	A directory that spans sites cannot be moved.
183	The list item being added already exists in the list (1).
206	The file or directory name is too long.
212	The database for the site collection is locked.
1150	Failed to update the list (1).
1359	An internal error occurred while moving a list item.
1638	The current schema version of the list (1) does not match the value in @CheckSchemaVersion.
1816	The site collection is over its allocated size quota.
8398	A directory could not be deleted.

The **proc_AddListItem** stored procedure MUST return no result sets.

3.1.5.5 proc_ChangeLevelForDoc

The **proc_ChangeLevelForDoc** stored procedure is invoked to update the publishing level of a document, indicating a change in its publishing status. Depending on the settings in effect on the containing list (1) and the permissions of the specified user making the request, **proc_ChangeLevelForDoc** also can update the document's Moderation Status (section 2.2.3.13), version number, and properties to reflect the change. If **@Level** is set to 255 (Checked Out) or 2 (Draft), and **@NewLevel** is set to 1 (Published), then the document's level value MUST be changed to 1 (Published). If **@Moderated** is 0, **@EnableMinorVersions** is 0, **@Level** is not set to 255 (Checked Out) and **@NewLevel** is not set to 1 (Published), then the document's level value MUST NOT change, and output parameters MUST NOT be changed. If **@Level** is 1 (Published) and **@NewLevel** is 2 (Draft), then the document's level value MUST be changed to 2 (Draft). If **@Level** is 255 (Checked Out) and the **@NewLevel** is 1 (Published), the document's **CheckOutUserId**, **CheckoutDate** and **CheckoutExpires** values MUST be set to NULL, and the document's container (document library or list (1)) value for **tp_CheckOutUserId** MUST be set to NULL.

```
PROCEDURE proc_ChangeLevelForDoc(
      @SiteId                       uniqueidentifier,
      @DirName                      nvarchar(256),
      @LeafName                     nvarchar(128),
      @Level                        tinyint           OUTPUT,
      @NewLevel                     tinyint,
      @ModerationStatus             int,
      @EnableMinorVersions          bit,
      @Moderated                    bit,
      @CreateVersion                bit,
      @UserId                       int,
      @Comment                      nvarchar(1023),
      @bUpdateModified              bit,
```


Copyright © 2014 Microsoft Corporation.

```
        @@DoclibRowId                   int                 OUTPUT,
        @@DocUIVersion                  int                 OUTPUT,
        @@DocFlagsOut                   int                 OUTPUT
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the document.

@DirName: The directory name of the document.

@LeafName: The leaf name of the document.

@Level: A publishing level value indicating the document's current publishing level. This output parameter also reflects the new level value for the document.

@NewLevel: A publishing level value indicating the requested level for the document.

@ModerationStatus: An integer value indicating the specified document's Moderation Status. See the Moderation Status section for a list of all valid values.

@EnableMinorVersions: A bit indicating whether minor versioning is enabled in the containing list (1). If the list (1) containing the specified document has minor versioning enabled, this MUST be set to 1; **otherwise**, this MUST be set to 0.

@Moderated**:** A bit indicating whether moderation is enabled on the containing list (1). If the list (1) containing the specified document has moderation enabled, this MUST be set to 1; otherwise, this MUST be set to 0.

@CreateVersion: A bit indicating whether versioning is enabled on the containing list (1). If the list (1) containing the specified document has versioning enabled, this MUST be set to 1; otherwise, this MUST be set to 0.

@UserId: A User Identifier (section 2.2.1.13) for the user requesting the document change. This value MUST refer to an existing User Identifier for the specified site collection.

@Comment: A text associated with the document on check in or publishing. When the document publishing level is updated from "Check Out" (255) to any other value, or from any value to "Publish" (1), **@Comment** can be set to a descriptive string or it can be NULL. In all other publishing level value transitions, this parameter MUST be NULL.

@bUpdateModified: A bit indicating if change to the date and time properties of the document is requested. If **@bUpdateModified** is set to 1, **proc_ChangeLevelForDoc** MUST update document date and time information in the store to note that the document was updated at the current time. Otherwise, the date and time values MUST be left unchanged.

@@DoclibRowId: An output parameter containing the value of the **DoclibRowId** column in the Docs View (section 2.2.7.4) for the document after the change.

@@DocUIVersion: An output parameter containing the UI version of the document after the change.

@@DocFlagsOut: An output parameter containing the Doc Flags (section 2.2.2.3) of the document after the change.

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

Copyright © 2014 Microsoft Corporation.

Value	Description
0	Successful execution.
87	The specified document could not be found, or the specified level is not the current level of the specified document.

The **proc_ChangeLevelForDoc** stored procedure MUST NOT return a result set.

3.1.5.6 proc_CheckoutDocument

The **proc_CheckoutDocument** stored procedure is invoked to place a short-term lock on a document, refresh or release an existing short-term lock, or to check out a document.

```
PROCEDURE proc_CheckoutDocument(
      @SiteId                       uniqueidentifier,
      @WebId                        uniqueidentifier,
      @DirName                      nvarchar(256),
      @LeafName                     nvarchar(128),
      @Level                        tinyint,
      @EnableMinorVersions          bit,
      @IsModerated                  bit,
      @UserId                       int,
      @CheckoutTimeout              int,
      @RefreshCheckout              bit,
      @CheckoutToLocal              bit,
      @IsForceCheckout              bit
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) for the site collection containing the document to be checked out, locked, or unlocked.

@WebId: The Site Identifier (section 2.2.1.11) for the site (2) containing the document.

@DirName: The directory name of the document.

@LeafName: The leaf name of the document.

@Level: A Publishing Level Type (section 2.2.2.6) value specifying the publishing status of the document.

@EnableMinorVersions: A bit flag specifying whether the document library or list (1) containing the document has minor version numbering enabled. If this parameter is set to "1", minor version numbering is enabled for the document library or list (1). This parameter MUST NOT be NULL.

@IsModerated: A bit flag specifying whether moderation is in effect on the document. If this parameter is set to "1", moderation is in effect on this document, which is used to implement an approval process to set the publishing level to published after the document is created or modified.

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

@CheckoutTimeout: Specifies the remaining time in minutes that short-term locking will be in effect for the document. A value of 0 means that the existing short-term lock MUST be released. The

Copyright © 2014 Microsoft Corporation.

@CheckoutTimeout parameter MUST be NULL if the document is being checked out instead of having a short-term lock applied.

@RefreshCheckout: A bit flag specifying whether the short-term lock on the document can be refreshed. If this parameter is set to "1", the existing short-term lock on the document MUST be refreshed for the number of seconds specified by the **@CheckoutTimeout** parameter. This parameter MUST be set to "0" to check out the document or to request a new short-term lock. 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 client for editing. If this parameter is set to "1", the user's client MUST make a local copy of the document stream for editing, and **proc_CheckoutDocument** MUST NOT make a checked-out version of the document in the store.

@IsForceCheckout: A bit flag specifying whether the containing document library requires documents to be checked out before any changes can be made. If this parameter is set to "1", the containing document library has 'Require Check Out' turned on.

Return Values**:** The **proc_CheckoutDocument** stored procedure returns an integer return code, which MUST be 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, or another user holds a short-term lock on the document.
154	Invalid minor version value. The minor version value for the document would exceed the maximum allowed value (511) if checked out.
158	Checkout required. The document is in a document library with the "Require Check Out" option set, but the document is not being checked out.
173	Checkout error. The document cannot be checked out, because it is already checked out to or locked by another user.
212	Site collection locked. The site collection is in disk write lock.
1630	Unsupported document type. The document specified is not valid for check out; folders and sites cannot be checked out.
1816	Disk quota error. The site collection disk quota has been reached.

The **proc_CheckoutDocument** stored procedure MUST return multiple result sets. Some of the result sets are returned 0 or more times depending upon conditions specified in the following sections, and all result sets that are returned will be sent in the order specified in the sections 3.1.5.6.1 through 3.1.5.6.5.

3.1.5.6.1 Link Info Single Doc Result Set

The Link Info Single Doc Result Set returns information about all forward and backward links associated with the document. The Link Info Single Doc Result Set MUST be returned once and MUST hold one row for each forward and backward link associated with the specified document at the specified publishing level.

Copyright © 2014 Microsoft Corporation.

The Link Info Single Doc Result Set is defined in the Common Result Sets Link Information Result Set (section 2.2.5.11).

3.1.5.6.2 Document Metadata Result Set

The Document Metadata Result Set returns the metadata for the document. The Document Metadata Result Set MUST be returned and MUST contain a single row corresponding to the checked-out or locked document.

The Document Metadata Result Set is defined in the Common Result Sets Document Metadata Result Set (section 2.2.5.6).

3.1.5.6.3 NULL Result Set

The NULL Result Set returns no data. The NULL Result Set MUST only be returned if the requested document version matching the **@**Level parameter exists for the specified user and the **@WebId** parameter is NULL. The NULL Result Set MUST return zero rows in a schema containing a single NULL column.

3.1.5.6.4 Event Receivers Result Set

The Event Receivers Result Set contains information about the list item event receivers defined for this document.

The Event Receivers Result Set MUST only be returned if the requested document matching the **@**Level parameter exists within the site (2) specified by the **@**WebId parameter for the current user. The Event Receivers Result Set MUST contain one row for each event receiver registered with an Event Host Type (section 2.2.3.5) of 3 (list item) for this document.

The Event Receivers Result Set is defined in the Common Result Sets Event Receivers Result Set (section 2.2.5.9).

3.1.5.6.5 Audit Mask Result Set

The Audit Mask Result Set returns audit configuration information. The Audit Mask Result Set MUST be returned with one row of audit configuration information on successful execution.

The Audit Mask Result Set is defined in the Common Result Sets Site Audit Mask Result Set (section 2.2.5.20).

3.1.5.7 proc_ClearLinks

The **proc_ClearLinks** stored procedure deletes all link information associated with a particular field (2) in a list item as part of a list item or document update. All link information that matches all of the parameters and does not have associated Web parts MUST be deleted by **proc_ClearLinks**. Link information associated with Web parts MUST NOT be deleted by **proc_ClearLinks**.

```
PROCEDURE proc_ClearLinks(
      @SiteId                       uniqueidentifier,
      @DirName                      nvarchar(256),
      @LeafName                     nvarchar(128),
      @Level                        tinyint,
      @FieldId                      uniqueidentifier)
);
```


Copyright © 2014 Microsoft Corporation.

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection that contains the list item with the link information to be deleted.

@DirName: The directory name of the list item with the link information to be deleted.

@LeafName: The leaf name of the list item with the link information to be deleted.

@Level: The Publishing Level Type (section 2.2.2.6) value associated with the list item with the link information to be deleted.

@FieldId: The field (2) identifier of the field in the list item with the link information to be deleted.

Return Values: The **proc_ClearLinks** stored procedure returns an integer return code, which MUST be 0, and MUST return no result sets.

3.1.5.8 proc_CreateDir

The **proc_CreateDir** stored procedure is invoked to create a directory or folder with a specified name at a specified location.

```
PROCEDURE proc_CreateDir(
      @DirSiteId                    uniqueidentifier,
      @DirWebId                     uniqueidentifier,
      @DirDirName                   nvarchar(256)                 OUTPUT,
      @DirLeafName                  nvarchar(128)                 OUTPUT,
      @DirLevel                     tinyint,
      @AddMinorVersion              bit,
      @DocFlags                     int,
      @CreateDirFlags               int,
      @UserId                       int               =NULL,
      @DirId                        uniqueidentifier  =NULL     OUTPUT,
      @ScopeId                      uniqueidentifier  =NULL     OUTPUT,
      @DoclibRowIdRequired          int               =NULL,
      @ScopeIdOverride              uniqueidentifier  =NULL,
      @bAlreadyExists               bit               =NULL     OUTPUT
);
```

@DirSiteId: The Site Collection Identifier for the site collection containing the directory to be created.

@DirWebId: The Site Identifier (section 2.2.1.11) for the site (2) containing the directory to be created.

@DirDirName: This is both an input and an output parameter. The input parameter MUST specify the URL of the parent location in which the specified directory is to be created. The directory name of the created directory MUST be returned as the output parameter value.

@DirLeafName: This is both an input and an output parameter. The input parameter MUST specify the name of the directory to be created in the parent location specified by the **@DirDirName** parameter. The leaf name of the created directory MUST be returned as the output parameter value.

@DirLevel: A Publishing Level Type (section 2.2.2.6) value describing the publishing status of the directory to be created. This value MUST be 1 (Published) if the directory is not being created in a list or document library. If this parameter is 2, specifying that the list item is to be created as a draft, the **@UserId** value MUST be set as the draft owner for the created directory.

@AddMinorVersion:

Copyright © 2014 Microsoft Corporation.

- If this parameter has a value of 1, **proc_CreateDir** MUST set the major version number of the created directory to 0 and the minor version number to 1.
- If this parameter has a value of 0, **proc_CreateDir** MUST set the major version number of the created directory to 1 and the minor version number to 0.

@DocFlags: A Doc Flags (section 2.2.2.3) value specifying metadata about the directory to be created. If the **@DocFlags** value has the 0x00002000 bit set (specifying a custom ordering of content types), this bit flag MUST be ignored and MUST NOT be included in the metadata associated with the created directory.

@CreateDirFlags: A 4-byte unsigned integer bit mask containing options used while creating the directory. The value MAY be 0, with none of the bits set, or bits can be set as follows.

Value	Description
0x00000007	These three bits contain an Attachment Flag value.
0x00000008	Return an access denied error if the specified directory already exists.
0x00000010	Do not promote the directory to a document library.
0x00000020	The directory contains a thicket.
0x00000380	These three bits contain a Moderation Status (section 2.2.3.13) value.
0xFFFFFC40	These values are currently unused and MUST be ignored.

@UserId: The User Identifier (section 2.2.1.13) for the current user who is directly or indirectly requesting the directory creation. This value can be NULL. If set, this MUST be used by **proc_CreateDir** for permission checking and setting ownership of the created directory. Otherwise, the ownership of the created directory MUST be set from the container.

@DirId: The identifier for the created directory. This is both an input and output parameter. If this parameter is passed in to **proc_CreateDir** with a non-NULL value, this value MUST be used as the document identifier (section 2.2.1.2) for the directory to be created. If this parameter is passed in with a NULL value, a new uniqueidentifier value MUST be generated for the directory. An existing directory MUST NOT have its document identifier (2) changed, regardless of the input value of this parameter. If **proc_CreateDir** creates a new directory, it MUST return the document identifier (2) for the created directory in the output parameter upon successful completion. If **proc_CreateDir** does not create a new directory (for example, if the directory already exists), this output parameter can contain an undefined value on return and MUST be ignored.

@ScopeId: An output parameter that MUST contain the non-NULL value specified for **@ScopeIdOverride** or, if a NULL value is specified for **@ScopeIdOverride**, the Scope Identifier (section 2.2.1.8) of the parent directory. This identifies the specific ACL to use for calculating the permission settings on the created directory.

@DoclibRowIdRequired: Specifies the List Item Identifier (section 2.2.1.6) to be used for this directory if created within a document library. If this parameter is not NULL and the directory to be created is within a document library, **proc_CreateDir** MUST set this as the List Item Identifier for the created directory. This parameter MUST be NULL for directories that are not created within a document library.

@ScopeIdOverride: Specifies the scope to use for the directory to be created. If this parameter is not NULL, it MUST be set as the Scope Identifier for the directory. Otherwise, the Scope Identifier of

Copyright © 2014 Microsoft Corporation.

the parent directory MUST be used. The scope set for the created directory MUST be returned in the **@ScopeId** output parameter.

@bAlreadyExists: An output parameter containing a bit flag set to 0 if the directory did not already exist and set to 1 if the directory already existed.

Return Values**:** The **proc_CreateDir** stored procedure MUST return an integer return code, which MUST be an SQL error code listed in the following table.

Value	Description
0	Successful execution.
3	The parent directory was not found.
5	The current user does not have sufficient permissions to create the directory at the specified location.
80	The specified directory already exists.
212	The site collection is locked.
1816	There is not enough database quota for the current user to complete the operation.

The **proc_CreateDir** stored procedure MUST return no result sets.

3.1.5.9 proc_DeleteAllDocumentVersions

The **proc_DeleteAllDocumentVersions** stored procedure is invoked to delete the specified draft (minor) versions or older published (major) versions of a document and optionally place them in the recycle bin.

```
PROCEDURE proc_DeleteAllDocumentVersions(
      @DocSiteId                    uniqueidentifier,
      @DocDirName                   nvarchar(256),
      @DocLeafName                  nvarchar(128),
      @UserId                       int,
      @DeleteOp                     int,
      @Level                        tinyint
);
```

@DocSiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the document whose specified versions are being deleted.

@DocDirName: The directory name of the document of which to delete the specified version. If there is no slash in the directory name, the value MUST be an empty string.

@DocLeafName: The leaf name of the document.

@UserId: The User Identifier (section 2.2.1.13) of the current user requesting the deletion. This value MUST refer to an existing User Identifier for the specified site collection.

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

Copyright © 2014 Microsoft Corporation.

Value	Description
3	The deleted document versions MUST NOT be placed in the recycle bin (nonrecoverable delete).
4	The deleted document versions MUST be placed in the recycle bin (recoverable delete).

@Level: A Publishing Level Type (section 2.2.2.6) value specifying which versions of the document or list item are to be deleted. The value MUST be one listed in the following table.

Value	Description
1	All versions of the document MUST be deleted, excluding the current version and the current published (major) version. If the current version and the current published (major) version are the same, then only one version will remain.
2	All draft (minor) versions of the document MUST be deleted, excluding the current version.

Return Values: The **proc_DeleteAllDocumentVersions** stored procedure returns an integer return code that MUST be listed in the following table.

Value	Description
0	Successful execution, or the document specified cannot be found in the site collection, and @DeleteOp was set to 3.
2	The document specified cannot be found in the site collection, and @DeleteOp was not set to 3.
1359	Invalid Parameter: @DeleteOp does not contain a valid value.

The **proc_DeleteAllDocumentVersions** stored procedure MUST NOT return a result set.

3.1.5.10 proc_DeleteDocBuildDependencySet

The **proc_DeleteDocBuildDependencySet** stored procedure is invoked to delete a **build dependency set** for a specified document.

```
PROCEDURE proc_DeleteDocBuildDependencySet(
      @DocSiteId                    uniqueidentifier,
      @DocDirName                   nvarchar(256),
      @DocLeafName                  nvarchar(128),
      @Level                        tinyint
);
```

@DocSiteId: The Site Collection Identifier (section 2.2.1.9) for a site collection containing the document.

@DocDirName: The directory name containing the document. If there is no slash in the directory name, the value MUST be an empty string.

@DocLeafName: The leaf name containing the document.

@Level: The Publishing Level Type value of the document.

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

Copyright © 2014 Microsoft Corporation.

The **proc_DeleteDocBuildDependencySet** stored procedure MUST NOT return a result set.

3.1.5.11 proc_DeleteDocumentVersion

The **proc_DeleteDocumentVersion** stored procedure is invoked to delete a document version and optionally place it in the recycle bin. A current version (whether published or draft) cannot be deleted.

```
PROCEDURE proc_DeleteDocumentVersion(
      @DocSiteId                    uniqueidentifier,
      @DocDirName                   nvarchar(256),
      @DocLeafName                  nvarchar(128),
      @DocVersion                   int,
      @UserId                       int,
      @DeleteOp                     int
);
```

@DocSiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the document whose specified version is being deleted.

@DocDirName: The directory name of the document to delete. If there is no slash in the directory name, the value MUST be an empty string.

@DocLeafName: The leaf name of the document to delete.

@DocVersion: The version of the document to delete, as seen in the WFE. This value MUST NOT be a current version.

@UserId: The User Identifier of the current user requesting the deletion. This value MUST refer to an existing User Identifier for the specified site collection.

@DeleteOp: A value determining 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 (nonrecoverable delete).
4	The deleted document versions MUST be placed in the recycle bin (recoverable delete).

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

Value	Description
0	Successful execution.
2	The version specified by @**DocSiteId**, @**DocDirName**, @**DocLeafName** and @**DocVersion** cannot be found.
186	Invalid Parameter: An invalid version was specified. The value specified in @**DocVersion** is a current version.

The **proc_DeleteDocumentVersion** stored procedure MUST NOT return a result set.

Copyright © 2014 Microsoft Corporation.

3.1.5.12 proc_DeleteUrl

The **proc_DeleteUrl** stored procedure is invoked to delete a document. **proc_DeleteUrl** accepts documents of all types except sites or attachment folders.

```
PROCEDURE proc_DeleteUrl(
      @WebSiteId                    uniqueidentifier,
      @WebId                        uniqueidentifier,
      @Url                          nvarchar(260),
      @UserId                       int,
      @LogChange                    bit               =1,
      @ListDeletedUrls              bit               =1,
      @ListDeletedAliases           bit               =0,
      @AttachmentsFlag              tinyint           =0,
      @AttachmentOp                 int               =3,
      @IgnoreCheckedOutFiles        bit               =0,
      @DeleteOp                     int               =3,
      @QueryAuditFlags              bit               =0,
      @FailedUrl                    nvarchar(260)     =NULL      OUTPUT,
      @DeleteTransactionId          uniqueidentifier  =
          '00000000-0000-0000-0000-000000000000'      OUTPUT,
      @eventData                    image             =NULL,
      @acl                          image             =NULL
);
```

@WebSiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the document.

@WebId: The Site Identifier (section 2.2.1.11) of the site (2) containing the document.

@Url: The store-relative form URL of the document.

@UserId: The User Identifier (section 2.2.1.13) for the current user requesting the operation. This parameter is used by **proc_DeleteUrl** for purposes of permission-checking. This value MUST refer to an existing User Identifier for the specified site collection.

@LogChange: If this bit is set to 1, an entry SHOULD be made in the **change log**. Otherwise, no entry SHOULD be made.

@ListDeletedUrls: If this bit is set to 1, it specifies that result set information about the deleted documents is requested. See the Deleted Documents Result Set (section 3.1.5.12.1).

@ListDeletedAliases: If this bit is set to 1, it specifies that result set information about deleted lists that had configured email addresses is requested. See the Deleted Aliased Lists Result Set (section 3.1.5.12.2).

@AttachmentsFlag: An Attachments Flag (section 2.2.3.1) describing the document specified by @Url.

@AttachmentOp: If this bit is set to 1, it specifies that on attachment deletion, the attachment folder record in the store MUST be updated to reflect whether it has any attachments remaining. Any other value specifies the stored procedure MUST refresh this information and other metadata in the store for the folder. This includes last modified date and time, the user identifier of the current user who made the modification, and incrementing of the internal folder version.

Copyright © 2014 Microsoft Corporation.

@IgnoreCheckedOutFiles: If this bit is set to 1, it specifies that the stored procedure MUST proceed with the delete operation on a folder even when it contains locked (short-term check-out) files.

@DeleteOp: This value defines the type of delete operation to attempt. If no value is specified, a default value of 3 is used. Otherwise, **@DeleteOp** MUST be one of the values listed in the following table.

Value	Description
3	Delete the item without placing the deleted item in the recycle bin.
4	Delete the item and place the deleted item in the recycle bin if enabled.
5	Delete the item and place the deleted item in the recycle bin.

@QueryAuditFlags: This parameter is set to 1 to specify that an audit entry MUST be created for the delete operation.

@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: If **proc_DeleteUrl** is called within an SQL transaction, @DeleteTransactionId MUST contain the transaction identifier. On output, this parameter MUST return either an all-zero GUID, indicating that no transaction was in effect, or a valid transaction ID, which MUST be used by the caller on subsequent calls.

@eventData: This parameter is reserved and MUST be ignored.

@acl: This parameter is reserved and MUST be ignored.

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

Value	Description
0	Successful execution.
3	The specified URL was not found.
5	The user is not authorized to make this change.
33	Cannot delete a folder containing checked-out or locked files.
50	Cannot delete a site (2) or an attachments folder.
51	Cannot delete a forms folder.
138	Cannot delete a URL containing lists.
161	Cannot delete a URL that contains sites.
206	The URL is too long.
1150	A concurrency violation or unknown error occurred.
1359	Internal error, or bad parameter specified. (Attachments Flag specifies an attachment file, but the URL to be deleted is a file in a document library.)

Copyright © 2014 Microsoft Corporation.

Value	Description
8398	There was an error deleting a list (1). At least one list (1) could not be deleted.

The **proc_DeleteUrl** stored procedure MUST return zero, one, or two result sets in the order shown.

3.1.5.12.1 Deleted Documents Result Set

The Deleted Documents Result Set contains information about the deleted documents. It MUST be returned on successful completion when **@ListDeletedUrls** is set to 1.

If the Deleted Documents Result Set is returned, it MUST contain one row for each document deleted.

If the document specified by **@Url** has a Document Store Type (section 2.2.2.4) of 0 (File), then the columns in the Deleted Documents Result Set MUST NOT be named.

```
{Url}                          nvarchar(260),
Type                          tinyint;
```

Url: The store-relative form URL of the deleted document.

Type: The Document Store Type of the deleted document.

3.1.5.12.2 Deleted Aliased Lists Result Set

The Deleted Aliased Lists Result Set contains information about all deleted lists that were configured with an email address. Such lists provide the implementation-specific email integration features. The Deleted Aliased Lists Result Set MUST be returned only when **@ListDeletedAliases** is set to 1 and the document specified by **@Url** does not have a Document Store Type of 0 (File).

If the Deleted Aliased Lists Result Set is returned, there MUST be one row returned for each deleted alias list.

```
{WebSiteId}                    uniqueidentifier,
tp_WebId                       uniqueidentifier,
tp_Id                          uniqueidentifier;
```

{WebSiteId}: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the deleted list (1).

tp_WebId: The Site Identifier (section 2.2.1.11) of the site (2) containing the list (1).

tp_Id: The List Identifier (section 2.2.1.5) of the deleted list (1).

3.1.5.12.3 Empty Deleted Aliased Lists Result Set

The Empty Deleted Aliased Lists Result Set MUST be returned only if **@ListDeletedAliases** is set to 1 and the document specified by **@Url** does not have a Document Store Type of 0 (File). Deletion of a file cannot cause the deletion of lists, and the Empty Deleted Aliased Lists Result Set is returned to indicate that the input parameter **@ListDeletedAliases** cannot be satisfied.

The Empty Deleted Aliased Lists Result Set has zero rows with a schema of two unnamed columns.

Copyright © 2014 Microsoft Corporation.

3.1.5.13 proc_DirtyDependents

The **proc_DirtyDependents** stored procedure is invoked to mark all items that depend on a given document, system setting, or status change, as "dirty," so that subsequent action can be taken to update them as necessary. In this context, a "dependent" item is an item that requires an update to its metadata when another item is modified. **proc_DirtyDependents** follows the full dependency chain until all dependent items are marked.

```
PROCEDURE proc_DirtyDependents(
      @SiteId                       uniqueidentifier,
      @DepType                      tinyint,
      @DepDesc                      nvarchar(270),
      @DepDescLike                  nvarchar(260)      =NULL)
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) for a site collection that contains the items with dependencies.

@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 2.2.2.3. The @**DepDesc** parameter is the metakey 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 (2) usage statistics. The @**DepDesc** parameter is the store-relative form URL of the site.

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

@DepDescLike: This parameter is unused and MUST be ignored.

Return Values: The **proc_DirtyDependents** stored procedure returns an integer return code, which MUST be one that is listed in the following table.

Value	Description
0	Successful execution.

The **proc_DirtyDependents** stored procedure returns no result sets.

3.1.5.14 proc_EnumLists

The **proc_EnumLists** stored procedure returns a list of the lists in a site (2), along with their associated metadata.

Copyright © 2014 Microsoft Corporation.

```
PROCEDURE proc_EnumLists (
      @WebId                        uniqueidentifier,
      @Collation                    nvarchar(32),
      @BaseType                     int               =NULL,
      @BaseType2                    int               =NULL,
      @BaseType3                    int               =NULL,
      @BaseType4                    int               =NULL,
      @ServerTemplate               int               =NULL,
      @FMobileDefaultViewUrl        bit               =NULL,
      @FRootFolder                  bit               =NULL,
      @ListFlags                    int               =NULL,
      @FAclInfo                     int               =NULL,
      @Scopes                       image             =NULL,
      @FRecycleBinInfo              bit               =NULL,
      @UserId                       int               =NULL
);
```

@WebId: The Site Identifier (section 2.2.1.11) for the site (2) containing the requested lists.

@Collation: A Windows Collation Name string identifier that follows the format for the T-SQL COLLATE clause. This MUST be the Collation Name of one of the valid Collation Order (section 2.2.3.4) values, with the case-insensitive and accent-sensitive flags set. For example, the default Collation Order is 25, Latin1_General which, with the case-insensitive and accent-sensitive flags set, has an SQL Collation Name string of "Latin1_General_CI_AS".

@BaseType: A parameter that restricts the returned lists by List Base Type (section 2.2.3.11). If this parameter is set to a valid List Base Type value, the lists returned MUST be restricted to lists with this List Base Type and with the List Base Types specified by **@**BaseType2, **@**BaseType3, and **@**BaseType4, if any. If this parameter is set to "-1" or NULL, the lists retrieved MUST NOT be restricted by List Base Type, and parameters **@**BaseType2, **@**BaseType3 and **@**BaseType4 MUST be ignored by **proc_EnumLists**.

@BaseType2, **@BaseType3**, **@BaseType4:** Additional parameters used to specify additional List Base Types for lists returned, depending on the value of the **@BaseType** parameter. If **@BaseType** is set to "-1" or NULL, these parameters MUST be ignored. If **@BaseType** is set to a valid List Base Type value, returned lists MUST be restricted to those whose List Base Type matches any specified by the four **@BaseType** parameters. Nonvalid and duplicated List Base Type values MUST have no effect on returned lists.

@ServerTemplate: The identifier for the List Server Template (section 2.2.3.12) that defines the base structure of the lists to be returned. If this parameter is set to a valid List Server Template value, returned Lists MUST be restricted to those whose List Server Template value matches **@ServerTemplate**. If this parameter is set to "-1" or NULL, this parameter MUST be ignored.

@FMobileDefaultViewUrl: A bit specifying whether information about the mobile default view URL is requested. If this parameter is set to "1", the information MUST be returned in the **tp_MobileDefaultViewUrl** column in the List Information Result Set. Otherwise, the **tp_MobileDefaultViewUrl** column MUST be returned with NULL values.

@FRootFolder: A bit specifying whether information about the root folder URL is requested. If this parameter is set to "1", the information MUST be returned in the **tp_RootFolder** column in List Information Result Set (section 3.1.5.14.1). Otherwise, the **tp_RootFolder** column MUST be returned with NULL values.

@ListFlags: A List Flags (section 2.2.2.5) value restricting the data returned by the List Information Result Set. If this parameter is not NULL, the stored procedure MUST only return data in the List

Copyright © 2014 Microsoft Corporation.

Information Result Set for lists with a **tp_Flags** value matching the **@ListFlags** value. Otherwise, this parameter MUST be ignored.

@FAclInfo: A flag specifying whether ACL information is requested. If this parameter is set to "1", the ACL information MUST be returned in the **tp_ACL** column in the List Information Result Set. Otherwise, the **tp_ACL** column MUST be returned with NULL values.

@Scopes: A binary image holding a concatenation of one or more Scope Identifiers (section 2.2.1.8), represented as 16-byte binary strings with no delimiters, each specifying a Scope Identifier. If this parameter is not NULL, this specifies that the stored procedure MUST only return data in the List Information Result Set for lists matching the specified scopes. Otherwise, this parameter MUST be ignored.

@FRecycleBinInfo: A bit specifying whether to return information about the contents of the implementation-specific recycle bin. If this parameter is set to "1", the Recycle Bin Information Result Set MUST be returned. Otherwise, the NULL Result Set (section 3.1.5.14.3) MUST be returned.

@UserId: A User Identifier (section 2.2.1.13) to specify the information to return in the Recycle Bin Information Result Set. If the **@**FRecycleBinInfo parameter is set to "1", the stored procedure MUST only return data for the specified user and list in the Recycle Bin Information Result Set. If the **@FRecycleBinInfo** parameter is not set to "1", this parameter MUST be ignored.

Return Values: The **proc_EnumLists** stored procedure returns an integer return code, which MUST be 0. **proc_EnumLists** MUST return two result sets in the order specified in the following sections.

3.1.5.14.1 List Information Result Set

The List Information Result Set returns information about Lists contained in the specified Site. The List Information Result Set MUST be returned and MUST contain zero or more rows, one for each list (1) that matches the specified input parameters.

```
tp_DocTemplateUrl             nvarchar(386),
tp_DefaultViewUrl             nvarchar(386),
tp_MobileDefaultViewUrl       nvarchar(256),
tp_Id                         uniqueidentifier,
tp_Title                      nvarchar(255),
tp_Description                ntext,
tp_ImageUrl                   nvarchar(255),
tp_Name                       nvarchar(38),
tp_BaseType                   int,
tp_FeatureId                  uniqueidentifier,
tp_ServerTemplate             int,
tp_Created                    datetime,
tp_Modified                   datetime,
tp_LastDeleted                datetime,
tp_Version                    int,
tp_Direction                  int,
tp_ThumbnailSize              int,
tp_WebImageWidth              int,
tp_WebImageHeight             int,
tp_Flags                      bigint,
tp_ItemCount                  int,
tp_AnonymousPermMask          bigint,
tp_RootFolder                 nvarchar(260),
tp_ReadSecurity               int,
```


Copyright © 2014 Microsoft Corporation.

```
tp_WriteSecurity              int,
tp_Author                     int,
tp_EventSinkAssembly          nvarchar(255),
tp_EventSinkClass             nvarchar(255),
tp_EventSinkData              nvarchar(255),
tp_EmailInsertsFolder         nvarchar(255),
tp_EmailAlias                 nvarchar(128),
tp_WebFullUrl                 nvarchar(256),
tp_WebId                      uniqueidentifier,
tp_SendtoLocation             nvarchar(512),
tp_ScopeId                    uniqueidentifier,
{tp_MaxMajorVersionCount}     int,
{tp_MaxMajorwithMinorVersionCount}int,
tp_DefaultWorkflowId          uniqueidentifier,
tp_HasInternalFGP             bit,
tp_ACL                        image;
```

tp_DocTemplateUrl: Contains the store-relative form URL of the document template associated with the list (1), if any, or NULL if none.

tp_DefaultViewUrl: Contains the store-relative form URL of the default view of the list (1). This value MUST NOT be NULL.

tp_MobileDefaultViewUrl: When **@FMobileDefaultViewUrl** is set to "1", this contains the store-relative form URL of the default view for mobile devices, if any, or NULL if none has been set. Otherwise, this MUST be NULL.

tp_Id: Contains the List Identifier (section 2.2.1.5) for the list (1).

tp_Title: Contains the display name, a short user-provided text description to identify the list (1).

tp_Description: Contains user-provided text describing the list (1).

tp_ImageUrl: Contains the store-relative form URL holding an image associated with the list (1).

tp_Name: Contains a string representation of the **tp_Id** GUID delimited by braces.

tp_BaseType: Contains the List Base Type value from which the list (1) is derived.

tp_FeatureId: Contains a Feature Identifier (section 2.2.1.4) for a feature associated with the list (1).

tp_ServerTemplate: Contains the value of the of the list template that defines the base structure of the list (1). This value can either be in the List Server Template (section 2.2.3.12) enumeration or can be a custom value as defined in the disk-based template (a template known to all web frontend clients) of the list (1). A custom list template value SHOULD be unique and more than 10000.

tp_Created: Contains a time stamp, in UTC format, specifying when the list (1) was created.

tp_Modified: Contains a time stamp, in UTC format, specifying when the list (1) was last modified.

tp_LastDeleted: Contains a time stamp, in UTC format, specifying when a list item was last deleted from the list (1). If no list item has been deleted, contains the value of **tp_Created**.

tp_Version: Contains an implementation-specific, internal version counter used in tracking modifications to the list's settings.

Copyright © 2014 Microsoft Corporation.

tp_Direction: Contains a value specifying the direction of text flow for WFE elements. Valid values are in the following table.

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

tp_ThumbnailSize: The width, in pixels, specified for use when creating thumbnail images of list items within this list (1). This value MUST be NULL for lists that do not have a List Server Template value of 109 (Image Library Template). Thumbnails and web images are WFE-generated images of documents and they are implementation-specific capabilities.

tp_WebImageWidth: The width, in pixels, specified for use when creating web images of list items within this list (1). This value MUST be NULL for lists that do not have a List Server Template value of 109 (Image Library Template).

tp_WebImageHeight: The height, in pixels, specified for use when creating web images of list items within this list (1). This value MUST be NULL for lists that do not have a List Server Template value of 109 (Image Library Template).

tp_Flags: Contains a List Flags (section 2.2.2.5) value describing list properties.

tp_ItemCount: Contains a count of list items in the list (1).

tp_AnonymousPermMask: Contains the WSS Rights Mask (section 2.2.2.13) that applies to an anonymous user in the scope of the list (1).

tp_RootFolder: When **@ FRootFolder** is set to "1", this contains the store-relative form URL of the root folder of the list (1). Otherwise, this MUST be NULL.

tp_ReadSecurity: Contains a value identifying the security policy for read access on list items. If this value is set to "1", users with read permissions can read all list items. Otherwise, users with read permissions can read only their own list items.

tp_WriteSecurity: Contains a value specifying the security policy in use for write access to list items. Valid values are 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.

tp_Author: Contains the User Identifier (section 2.2.1.13) of the user who created the list (1).

tp_EventSinkAssembly: Contains the assembly name of the event sink handler if an event sink handler is registered for the list (1) or NULL if none exists.

tp_EventSinkClass: Contains the assembly class identifier of the event sink handler if an event sink handler is registered for the list (1) or NULL if none exists.

Copyright © 2014 Microsoft Corporation.

tp_EventSinkData: Contains Unicode string data specific to the implementation of the event sink associated with this list (1) or NULL if none exists.

tp_EmailInsertsFolder: Contains a store-relative form URL specifying the directory on the configured email inserts server that SHOULD be inspected for new email messages to be processed for this list (1) MAY be NULL if the list (1) is not email-enabled.

tp_EmailAlias: Contains the email alias of the list (1). This alias is used to allow files to be sent directly to the list (1) through an implementation-specific email handling feature. MAY be NULL if the list (1) is not email-enabled.

tp_WebFullUrl: Contains the store-relative form URL of the site (2) that contains the list (1).

tp_WebId: Contains the Site Identifier (section 2.2.1.11) of the site (2) that contains the list (1).

tp_SendToLocation: Contains an implementation-specific string of the URL used in the "Send To Location" feature. MAY be NULL if the list (1) is not email-enabled or if the list (1) is not configured with a "Send To Location".

tp_ScopeId: Contains the Scope Identifier (section 2.2.1.8) of the scope for the list (1).

{tp_MaxMajorVersionCount}: Contains the number of major versions that will be retained for this document if versioning is enabled.

{tp_MaxMajorwithMinorVersionCount}: Contains the number of major versions that will have their associated minor versions retained for this document if versioning is enabled.

tp_DefaultWorkflowId: Contains the Workflow Identifier (section 2.2.1.16) of the default workflow (2) associated with the list (1) or NULL if none exists.

tp_HasInternalFGP: This flag is set to 1 if there have ever been list items that have had a unique ACL applied. Otherwise, this flag is set to 0.

tp_ACL: When **@FAclInfo** is set to "1", this MUST contain a binary image of the ACL for this list (1) if one is defined; otherwise, it MUST be NULL.

3.1.5.14.2 Recycle Bin Information Result Set

The Recycle Bin Information Result Set returns information about list items from the specified list that the specified user marked as deleted in the recycle bin.

If **@FRecycleBinInfo** is "1", the Recycle Bin Information Result Set MUST be returned and MUST contain a single row with two columns specified as follows.

```
{RecycleBinCount}              int,
{RecycleBinSize}               int;
```

{RecycleBinCount}: Contains a count of the number of items in the recycle bin matching the specified user and list (1).

{RecycleBinSize}: Contains the total size, in bytes, of items in the recycle bin matching the specified user and list (1).

Copyright © 2014 Microsoft Corporation.

3.1.5.14.3 NULL Result Set

The NULL Result Set is an empty placeholder returned when recycle bin information is not requested. The NULL Result Set MUST only be returned if **@FRecycleBinInfo** is not 1 and MUST contain no rows, in a schema consisting of a single, unnamed NULL column.

3.1.5.15 proc_FetchDocForHttpGet

The **proc_FetchDocForHttpGet** stored procedure is invoked to fetch a document for the HTTP GET & HEAD operations and to provide information necessary to render the document on the WFE. Different sets of information are provided depending on the type of request (HEAD or GET) or the type document requested (such as a file, a web page, or a list item view webpage).

```
PROCEDURE proc_FetchDocForHttpGet(
      @DocSiteId                    uniqueidentifier,
      @DocDirName                   nvarchar(256),
      @DocLeafName                  nvarchar(128),
      @LooksLikeAttachmentFile      bit,
      @IfModifiedSince              datetime,
      @FetchType                    int,
      @ValidationType               int,
      @ClientVersion                int,
      @ClientId                     uniqueidentifier,
      @PageView                     tinyint,
      @FetchBuildDependencySet      bit,
      @SystemID                     varbinary(512),
      @CurrentVirusVendorID         int,
      @PrefetchListScope            bit,
      @ChunkSize                    int,
      @DGCacheVersion               bigint,
      @MaxCheckinLevel              tinyint,
      @HonorLevel                   bit,
      @CurrentFolderUrl             nvarchar(260),
      @Level                        tinyint           OUTPUT
);
```

@DocSiteId: The Site Collection Identifier (section 2.2.1.9) for a site collection containing the document.

@DocDirName: The directory name of the document.

@DocLeafName: The leaf name of the requested document.

@LooksLikeAttachmentFile: Set to 1 if the URL string indicates that the document is an attachment to a list item or an attachment folder; otherwise, set to 0.

@IfModifiedSince: If the caller has a cached copy of the document, **@IfModifiedSince** indicates the Last Modified time value for the cached copy. Otherwise, **@IfModifiedSince** MUST be NULL. Used in combination with **@ValidationType** to determine whether the document contents will be returned.

@FetchType: Indicates the type of request. This parameter MUST be set to 0 to specify information is requested to satisfy a normal HTTP GET, including the content of the document. This parameter MUST be set to 1 if information is needed only to satisfy an HTTP HEAD request, which does not include the content of the document. All nonzero values MUST also be treated as 1.

Copyright © 2014 Microsoft Corporation.

@ValidationType: Specifies the criteria used to determine whether the document contents return. This parameter is used for cache optimization. The only valid values of the **@ValidationType** bits are specified as follows.

Value	Description
0	None: The WFE has no cached copy, and document contents are requested unconditionally.
1	E-Tag: Return document content if the WFE version specified by input parameter **@ClientVersion**) does not match the document version in the BEDS, or if the document identifier (2) (section 2.2.1.2) specified by input parameter **@ClientId** does not match the document identifier (2) in the BEDS.
2	Last modified time: Return document content if the Last Modified time specified by the input parameter **@IfModifiedSince**) is earlier than the Last Modified time of the document in the BEDS.
3	Both E-Tag and Last modified time: Return document content if the WFE version specified by input parameter **@ClientVersion** does not match the document version in the BEDS, or if the document identifier (2) specified by input parameter **@ClientId** does not match the document identifier (2) in the BEDS, or if the Last Modified time specified by the input parameter **@IfModifiedSince** is earlier than the Last Modified time of the document in the BEDS.

@ClientVersion: Version of the document as seen in the WFE. Used for client cache validation, as indicated by the value of **@ValidationType**.

@ClientId: The document identifier (2) as seen in the WFE. Used for client cache validation in combination as indicated by the value of **@ValidationType**.

@PageView: Non-NULL values indicate that the document is a view webpage, an implementation-specific webpage that renders an item or items in a list (1) or document library. This also indicates that information is needed to enable WFE rendering based on metadata about the view page, the item or items being rendered, the user's browsing context, the overall site's navigation scheme, and the user's security privileges. Information requested includes site (2) metadata, containing list metadata, users and site groups metadata, Web Parts, and related list metadata. In addition, a value of 1 indicates that the metadata SHOULD be security trimmed to the user specified by **@SystemId**. The value 0 or other non-NULL values indicate that information is requested as seen by all users. A NULL value indicates that the document is not a view webpage.

@FetchBuildDependencySet: This parameter is set to 1 to indicate that information is requested about dependency documents. This is used when the document is dynamically built from a set of objects, such as a master page, and a refresh of the set is needed to generate an up-to-date rendering of the document.

@SystemID: The **SystemID** (section 2.2.1.12) of the user originating the request. This parameter MUST refer to a valid, nondeleted user in the specified site collection, or be set to NULL to indicate the anonymous user. Used to determine access privileges.

@CurrentVirusVendorID: Identifier of the anti-virus vendor, if any, registered for the document, as seen by the WFE. May be NULL.

@PrefetchListScope: This parameter is set to 1 to indicate that containing list scope metadata is requested.

@ChunkSize: Specifies the size, in bytes, that the document MUST be in order for the document's content to be returned as part of this query. If the document is larger than this value, a single 0

Copyright © 2014 Microsoft Corporation.

byte is returned, and the WFE can request the remainder of the document in a subsequent operation.

@DGCacheVersion: The version number of the Domain Group Map Cache in the WFE. This can be compared with the Domain Group Map Cache version number on the BEDS returned in the Domain Group Cache Versions Result Set to determine whether updates are needed. A **@DGCacheVersion** value of -2 specifies that information about the Domain Group Map Cache is not requested, and the Domain Group Cache BEDS Update Result Set and the Domain Group Cache WFE Update Result Set MUST NOT be returned.

@MaxCheckinLevel: If **@HonorLevel**=1, **@MaxCheckinLevel** specifies the document publishing level requested. Otherwise, this parameter is ignored.

@HonorLevel: If set to 1, fetch the document with the level specified in **@MaxCheckinLevel**. Otherwise, fetch the maximum level available to the user specified by **@SystemId**.

@CurrentFolderUrl: The URL of the folder that the user was browsing before the specified document was requested. This is used to preserve the browsing context when the fetch request is redirected to a view page. When fetching an item in a List or a file in a document library, an attachment to an item, a folder, or a list view, the specified document will be a view page, and **@CurrentFolderUrl** will be used to render the browsing context as appropriate.

@Level**:** An output parameter specifying the publishing level value of the document.

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

Value	Description
0	Successful execution.
2	Document not found: The document specified by **@DocSiteId**, **@DocDirName**, **@DocLeafName** (and **@MaxCheckinLevel**, if specified) was not found, or the document is an attachment folder; or the document is a folder, but no welcome page, home page, view webpage, or other redirect webpage was found for it.
18	Successful execution, file not modified: Based on input parameters, the document's copy in the client cache is valid and can be used.
146	The document does not exist, but a welcome page was found, as specified by the Welcome Page Redirect Information Result Set.
1168	The site collection specified by **@DocSiteId** was not found.
1271	The site (2) is read/write locked.

The **proc_FetchDocForHttpGet** stored procedure MUST return zero or more result sets upon successful execution. If the document is not found, zero result sets MUST be returned. The following result sets MUST be returned only if the document has a Document Store Type of 0 (File)**:**

Site Collection Audit Mask Result Set

List Audit Mask Result Set

Document Build Dependency Set Result Set

Document Build Dependency Metadata Result Set

Copyright © 2014 Microsoft Corporation.

Domain Group Cache Versions Result Set

Site Metadata Result Set

Event Receivers Result Set

WebParts Metadata, Personalized Result Set

WebParts Metadata, Nonpersonalized Result Set

ListMetadata Result Set

List Event Receivers Result Set

List Security Information Result Set

List Web Parts Result Set

Content Type Order Result Set

Current Folder Scope Result Set

Site Features List Result Set

Navigation Context Security Information Result Set

NULL Navigation Context Security Information Result Set

Empty Navigation Context Security Information Result Set

Other **proc_FetchDocForHttpGet** result sets return conditionally, as described in each result set section.

3.1.5.15.1 HTTP Document Metadata Result Set

The HTTP Document Metadata Result Set returns the core document metadata, including the effective ACL and anonymous permission mask. This Result Set MUST be returned if the specified document exists.

```
{Size}                      int,
{DocFlags}                  int,
{FullUrl}                   nvarchar(260),
{WebId}                     uniqueidentifier,
{FirstUniqueWebId}          uniqueidentifier,
{SecurityProvider}          uniqueidentifier,
{Dirty}                     bit,
{TimeLastWritten}           datetime,
{CharSet}                   int,
{Version}                   int,
{DocId}                     uniqueidentifier,
{LeafName}                  nvarchar(128),
InDocLibrary                bit,
IsAttachment                bit,
NeedManageListRight         int,
{SiteFlags}                 int,
Acl                         image,
AnonymousPermMask           bigint,
{ListIdForPermissionCheck}  uniqueidentifier,
```


Copyright © 2014 Microsoft Corporation.

```
{PermCheckedAgainstUniqueList} int,
DraftOwnerId                   int,
ListFlags                      bigint,
Level                          tinyint,
{IsCurrentVersion}             bit,
{Type}                         tinyint,
{VirusVendorID}                int,
{VirusStatus}                  int,
{VirusInfo}                    nvarchar(255),
{ContentModifiedSince}         bit,
{ProgId}                       nvarchar(255),
{DoclibRowId}                  int,
{Language}                     int,
{DirName}                      nvarchar(256);
```

{Size}: The size, in bytes, of the document.

{DocFlags}: A Doc Flags (section 2.2.2.3) value describing the document.

{FullUrl}: The complete store-relative form URL for the requested item.

{WebId}: The Site Identifier (section 2.2.1.11) of the site (2) containing the document.

{FirstUniqueWebId}: The Site Identifier of the site (2) whose security permissions are effective for the site containing the specified document.

{SecurityProvider}: COM CLSID of the security provider (1) for this site (2). This MUST be NULL for sites using the native security implementation.

{Dirty}: A bit declaring that this document MUST have dependency update processing performed before its stream is returned to an external agent. If this document does not have a content Stream, the value is implementation-dependent and MUST be ignored.

{TimeLastWritten}: The time, in UTC format, when the document content was last modified. This does not necessarily correspond to the actual time when the document was last saved.

{CharSet}: An character set associated with the document. Any windows code page identifier is valid for CharSet. May be NULL.

{Version}: An integer incremented any time a change is made to this document, used for internal conflict detection.

{DocId}: The document identifier (section 2.2.1.2) of the requested document.

{LeafName}: The leaf name of the requested document.

InDocLibrary: If 1, the document is in a document library; otherwise, it is not.

IsAttachment: If 1, the document is an attachment; otherwise, it is not.

NeedManageListRight: If 1, then the user is required to have the "Manage List" right in order to read the document; otherwise, the user is not.

{SiteFlags}: A Site Collection Flags (section 2.2.2.9) value describing the configuration of the site collection containing the document.

Copyright © 2014 Microsoft Corporation.

Acl: The binary serialization of the WSS ACL Format (section 2.2.4.6) ACL in effect for the document.

AnonymousPermMask: A WSS Rights Mask (section 2.2.2.13) that indicates the rights granted to a user that is anonymous, or has no specific rights, to this document.

{ListIdForPermissionCheck}: The identifier for the list (1) containing the document. MUST be NULL if this document is not in a list (1).

{PermCheckedAgainstUniqueList}: MUST be 0.

DraftOwnerId: The identifier for the user who published the document as a draft. This value is non-NULL only if the requested document is a draft version.

ListFlags: A List Flags (section 2.2.2.5) value describing the list (1) that contains the document. If the document is not in a list (1), the value MUST be 0.

Level: A publishing level value specifying the publishing status of this document.

{IsCurrentVersion}: If set to 1, this is a current version of the document.

{Type}: The Document Store Type (section 2.2.2.4) of this item.

{VirusVendorID}: The identifier of the virus scanner that processed this document. This value MUST be NULL if the document has not been processed by a virus scanner.

{VirusStatus}: The Virus Status (section 2.2.3.18) of the document. This value MUST be NULL if the requested document does not exist or if it has not been processed by a virus scanner.

{VirusInfo}: A string containing a provider-specific message returned by the virus scanner when it last processed the document.

{ContentModifiedSince}: A bit indicating whether the document has been modified, depending on the validation type requested. Set to 1 if any of the following are true**:** the document is a dynamic document type; the document requires a dependency update; validation type is "None" (0); validation type is "E-tag" (1) and the value of **@ClientVersion** disagrees with the document version in the store, or the value of **@ClientId** disagrees with the document identifier (2) in the store; validation type is "Last modified" (2) and the last modification date of the document's copy in the store is more recent than specified in **@IfModifiedSince**. In all other cases, **ContentModifiedSince** is set to 0.

{ProgId}: Designates a preferred application to open the document. The **ProgId** is used to distinguish between different applications that save files with a given file extension (for example, different editors for HTML or XML files). This value MUST be NULL if the parser did not specify a **ProgId** when the document was saved.

{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 WFE. This parameter can be NULL.

{DirName}: The directory name of the requested document.

Copyright © 2014 Microsoft Corporation.

3.1.5.15.2 Domain Group Cache Versions Result Set (1)

The Domain Group Cache Versions Result Set contains information about the version numbers associated with the Domain Group Map Caches on the WFE and on the BEDS for the specified site collection.

The Domain Group Cache Versions Result Set MUST be returned if the specified document exists and MUST contain one row of version number data. If the specified **@DGCacheVersion** value is -2, all columns returned MUST have the value -2, indicating that the value MUST NOT be used for comparison. The Domain Group Cache Versions Result Set is defined in the Common Result Sets Domain Group Cache Versions Result Set (section 2.2.5.4).

3.1.5.15.3 Domain Group Cache BEDS Update Result Set

The Domain Group Cache BEDS Update Result Set contains information to be used in recomputing the Domain Group Map Cache.

The Domain Group Cache BEDS Update Result Set MUST be returned if **@DGCacheVersion** is not -2 ('Skip') and the real Domain Group Map Cache version is more recent than the cached version on the BEDS; that is, if the value of RealVersion is greater than the value of CachedVersion in the Domain Group Cache Versions Result Set (section 2.2.5.4).

If the Domain Group Cache BEDS Update Result Set is returned, it indicates that the copy of the Domain Group Map Cache on the BEDS is out of date and MUST be recomputed to ensure that proper security checks can be made.

When returned, the Domain Group Cache BEDS Update Result Set MUST have a single row. The Domain Group Cache BEDS Update Result Set is defined in the Common Result Sets Domain Group Cache BEDS Update Result Set (section 2.2.5.3).

3.1.5.15.4 Domain Group Cache WFE Update Result Set

The Domain Group Cache WFE Update Result Set contains the binary data needed to refresh the domain group map cache.

The Domain Group Cache WFE Update Result Set MUST be returned only if **@DGCacheVersion** is not -2 ('Skip') and the cached version on the BEDS is up-to-date; that is, if the value of RealVersion is not greater than the value of CachedVersion in the Domain Group Cache BEDS Update Result Set (section 2.2.5.3).

The Domain Group Cache WFE Update Result Set is defined in the Common Result Sets Domain Group Cache WFE Update Result Set (section 2.2.5.5).

3.1.5.15.5 User Information Result Set

The User Information Result Set returns information about the user specified in **@**UserId.

```
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,
```


Copyright © 2014 Microsoft Corporation.

```
UserId                          int,
SiteSecurityVersion             bigint;
```

tp_Id: The identifier of the user.

tp_SiteAdmin: Indicates whether the specified user is an administrator on the site collection specified in **@DocSiteId**.

tp_IsActive: Set to "1" if the specified user is an active user in the site collection specified in **@DocSiteId**.

tp_Login: The login name of the specified user.

tp_Email: The email 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 UTC format, when the External Group Token (section 2.2.4.2) for the specified user was last updated.

tp_Token: A WSS User Token (section 2.2.4.9) value specifying the site group membership of the specified user.

UserId: The site membership identifier of the specified user. This parameter can be NULL if the user has not been added as a member to the site (2) whose permissions are in effect on the document.

SiteSecurityVersion: The current security information version of the site collection containing this document.

3.1.5.15.6 Welcome Page Redirect Information Result Set

The Welcome Page Redirect Information Result Set returns if the document is a site (2) or a folder with a configured welcome page, or the welcome page itself. The Welcome Page Redirect Information Result Set contains information about the welcome page.

```
{RedirectType}                  tinyint,
{RedirectUrl}                   nvarchar(260),
WelcomePageParameters           ntext,
{ContentTypeId}                 varbinary(512);
```

{RedirectType}: The Redirect Type (section 2.2.3.15) of the URL. MUST be 0, indicating a welcome page redirect URL.

{RedirectUrl}: The URL of the welcome page.

WelcomePageParameters: This MUST contain any URL parameters configured for the welcome page. This value can contain a query string starting with "?" or a hash parameter starting with "#".

{ContentTypeId}: This value MUST be NULL.

Copyright © 2014 Microsoft Corporation.

3.1.5.15.7 Non-Welcome Page Redirect Information Result Set

The Non-Welcome Page Redirect Information Result Set returns if the document is a site (2) or a folder, where a welcome page is not configured. Depending on the specified document URL, the redirect can be to a home page, a list view web page, or to a provisioning page URL. The Non-Welcome Page Redirect Information Result Set contains information about the redirect URL. The Non-Welcome Page Redirect Information Result Set MUST return a single row.

```
{RedirectType}                tinyint,
{RedirectUrl}                 nvarchar(260),
{WelcomePageParameters}       ntext,
{ContentTypeId}               varbinary(512);
```

{RedirectType}: The Redirect Type of the URL. This parameter MUST NOT be 0 (welcome page.) For all other valid values, see the Redirect Type (section 2.2.3.15).

{RedirectUrl}: The full redirect URL.

{WelcomePageParameters}: MUST be NULL.

{ContentTypeId}: A binary identifier of the document's content type.

3.1.5.15.8 Document Content Stream Result Set

The Document Content Stream Result Set contains the document's binary stream and associated metadata. It MUST return only if **@FetchType** is NOT set to 1 (not indicating an HTTP HEAD-only request).

The Document Content Stream Result Set MUST return zero or one rows. If the document is modified, a single row MUST return. A document is considered modified subject to the semantics indicated by the input parameter **@ValidationType**, or if its Virus Vendor ID has been updated since the client last retrieved it. In such a case, the return code 18 MUST be returned upon successful completion of **proc_FetchDocForHttpGe**t. Otherwise, zero rows MUST return.

```
{Content}                     image,
{Size}                        int,
{SetupPathVersion}            tinyint,
{SetupPath}                   nvarchar(255),
{SetupPathUser}               nvarchar(255),
{Dirty}                       bit,
{Version}                     int,
{Id}                          uniqueidentifier;
```

{Content}: The document's content stream. For a ghosted document, this MUST be NULL. Otherwise, if the content is larger than the value specified in the **@**ChunkSize parameter, a single zero byte will be returned, and the WFE can request individual chunks of content in a subsequent request.

{Size}: The size of the document, in bytes.

{SetupPathVersion}: For a ghosted document, this parameter governs the setup path location to which the **SetupPath** fragment is relative. This value MUST be NULL if the document does not exist, and it is undefined for a document that was never ghosted. The following are all possible valid values.

Copyright © 2014 Microsoft Corporation.

Value	Description
2	Relative to the install location of Windows SharePoint Services 2.0 on the WFE (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 WFE (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\12).

{SetupPath}: For a document that has ever been ghosted, 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.

{SetupPathUser}: If the document has ever been ghosted, this contains the login name of the user who created the ghosted document. Otherwise, this parameter MUST be NULL.

{Dirty}: This parameter is set to 1 if this document has had "content stream needs dependency update" processing performed; otherwise, it is set to 0. If the document does not have a content stream, the value is implementation-dependent and MUST be ignored.

{Version}: An integer value tracking the document version in a linearly increasing, implementation-specific version numbering system.

{Id}: The Document Identifier (section 2.2.1.2) of the document.

3.1.5.15.9 Site Collection Audit Mask Result Set

The Site Collection Audit Mask Result Set contains the information about the Audit Flags (section 2.2.2.1) associated with the site collection containing the specified document.

The Site Collection Audit Mask Result Set is defined in the Common Result Sets Site Audit Mask Result Set (section 2.2.5.20).

3.1.5.15.10 List Audit Mask Result Set

The List Audit Mask Result Set contains information about the Audit Flags (section 2.2.2.1) associated with the list (1) containing the document.

The List Audit Mask Result Set MUST return one row if the document is contained in a list (1); otherwise, zero rows are returned.

```
tp_Id                       uniqueidentifier,
tp_AuditFlags               int,
tp_InheritAuditFlags        int,
{GlobalAuditMask}           int,
{URL}                       nvarchar(516)
```

tp_Id: The List Identifier (section 2.2.1.5) of the list (1) containing the document.

tp_AuditFlags: An Audit Flags value determining the operations to be tracked on the list (1).

tp_InheritAuditFlags: An Audit Flags value determining the operations to be tracked on the document as inherited from the document's container.

{GlobalAuditMask}: An Audit Flags value determining the operations to be tracked across the site collection that contains the document.

Copyright © 2014 Microsoft Corporation.

{URL}: The URL of the list (1) containing the document.

3.1.5.15.11 Document Build Dependency Set Result Set

The Document Build Dependency Set Result Set returns a binary image containing implementation-specific dependency information for a published document (build dependency set). The Document Build Dependency Set Result Set MUST return only when requested (**@**FetchBuildDependencySet is set to 1) and if the document is published.

```
BuildDependencySet            image;
```

BuildDependencySet**:** A binary image holding implementation-specific information about dependent documents. NULL indicates that there is no information about dependencies. A BuildDependencySet of size 0 indicates a document with no dependencies.

3.1.5.15.12 Document Build Dependency Metadata Result Set

The Document Build Dependency Metadata Result Set contains information about each document in the specified document's build dependency set. The Document Build Dependency Metadata Result Set MUST return only if the build dependency set was requested and is not empty (that is, **@FetchBuildDependencySet** is set to 1, and the image column returned in the Document Build Dependency Set Result Set (section 3.1.5.15.11) is not NULL or zero-length).

```
DirName                    nvarchar(256),
LeafName                   nvarchar(128),
WebsFullUrl                nvarchar(256),
FirstUniqueAncestorWebId   uniqueidentifier,
SecurityProvider           uniqueidentifier,
Acl                        image,
AnonymousPermMask          bigint,
WebId                      uniqueidentifier,
ListId                     uniqueidentifier,
UniqueList                 bit,
InDocLibrary               bit,
DraftOwnerId               int,
ListFlags                  bigint,
ProgId                     nvarchar(255),
SetupPathVersion           tinyint,
SetupPath                  nvarchar(255),
TimeLastWritten            datetime,
MasterUrl                  nvarchar(260),
CustomMasterUrl            nvarchar(260),
Version                    int,
Id                         uniqueidentifier,
BuildDependencySet         image;
```

DirName: The directory name of the document.

LeafName: The leaf name of the document.

WebsFullUrl: The store-relative form URL of the document.

FirstUniqueAncestorWebId: The GUID identifier of the closest site (2) in this site's ancestor chain that does not inherit security settings from its parent site.

Copyright © 2014 Microsoft Corporation.

SecurityProvider: COM CLSID of the external security provider (1) for this site (2). This is NULL for sites using the native security implementation.

Acl: The binary serialization of the WSS ACL Format (section 2.2.4.6) ACL for this site (2). This is either explicitly defined or inherited from the parent object of the site (2).

AnonymousPermMask: A WSS Rights Mask (section 2.2.2.13) that indicates the rights granted to a user that is anonymous, or has no specific rights, to the document.

WebId: The GUID for a site (2) containing the document.

ListId: The GUID for the list (1) containing the document.

UniqueList: This value MUST be zero. It can be ignored.

InDocLibrary: If 1, the document is contained within a document library.

DraftOwnerId: The user identifier for the owner of the last checked-in draft version of the document.

ListFlags: A List Flags value describing the list (1) that contains this document. This value MUST be NULL if the document is not stored in a list (1).

ProgId: Designates a preferred application to open this document. The **progid** is used to distinguish between different applications that save files with a given file extension (that is, different editors for HTML or XML files). This value MUST be NULL if the document does not exist or the parser did not specify a **progid** when the document was saved.

SetupPathVersion: For a ghosted document, this governs the setup path location to which the **SetupPath** fragment is relative. This value MUST be NULL if the requested document does not exist and is undefined for documents that were never ghosted. The following are values are valid.

Value	Description
2	The SetupPath is relative to the install location of Windows SharePoint Services 2.0 on the WFE (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\60).
3	The SetupPath is relative to the install location of Windows SharePoint Services 3.0 on the WFE (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\12).

SetupPath: For a document that is now or once was ghosted, this contains the setup path fragment relative to the base setup path described earlier by the **SetupPathVersion** value, where the content stream of this document can be found. This value MUST be NULL in the case of a document that does not exist or a document that was never ghosted.

TimeLastWritten: A time stamp, in UTC format, specifying when any changes were made to the document stream. Does not reflect changes made strictly to the MetaInfo or other item properties.

MasterUrl: The URL for the master page registered on the site (2) for use in pages of the site when rendered on the WFE.

CustomMasterUrl: The URL for an alternate master page registered on the site (2) for use in pages of the site rendered on the WFE.

Version: A counter incremented any time a change is made to this document and used for internal conflict detection.

Id: The Document Identifier (section 2.2.1.2).

Copyright © 2014 Microsoft Corporation.

BuildDependencySet: An image value specifying further document dependencies.

3.1.5.15.13 Domain Group Cache Versions Result Set (2)

This is the second instance of the Domain Group Cache Versions Result Set for **proc_FetchDocForHttpGet**.

This instance of the Domain Group Cache Versions Result Set MUST be returned only if the input parameter **@PageView** is not NULL, as part of a series of result sets describing view webpage document metadata. This instance is redundant and can be ignored. The value of all columns returned in this result set MUST be -2, indicating that the value MUST NOT be used for comparison. The Domain Group Cache Versions Result Set is defined in the Common Result Sets Domain Group Cache Versions Result Set (section 2.2.5.4).

3.1.5.15.14 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 view web page document metadata.

The Site Metadata Result Set is defined in the Common Result Sets Site Metadata Result Set (section 2.2.5.23).

3.1.5.15.15 Event Receivers Result Set

The Event Receivers Result Set contains information about the event receivers defined for the site (2) containing the specified document.

The Event Receivers Result Set is part of a series of result sets describing view webpage-related metadata. It MUST return ONLY if the input parameter **@PageView** is not NULL, and it will NOT return if **proc_FetchDocForHttpGe**t 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 (2). The Event Receivers Result Set is defined in the Common Result Sets Event Receivers Result Set (section 2.2.5.9).

3.1.5.15.16 WebParts Metadata, Personalized Result Set

The Web Parts Metadata, Personalized Result Set contains the core metadata about the Web parts appearing on the specified document, personalized for the user specified in **@UserId**.

The Web Parts Metadata, Personalized Result Set is part of a series of result sets describing view Web page-related metadata. It MUST return ONLY if the input **@PageView** is NOT 0.

The Web Parts Metadata, Personalized Result Set will NOT return if **proc_FetchDocForHttpGe**t returns a code of 1271, indicating that the site collection was locked.

The Web Parts Metadata, Personalized Result Set MUST contain one row per Web part.

```
tp_Id                       uniqueidentifier,
{tp_ZoneID}                 nvarchar(64),
tp_WebPartTypeId            uniqueidentifier,
{tp_IsIncluded}             bit,
{tp_FrameState}             tinyint,
tp_AllUsersProperties       image,
```


Copyright © 2014 Microsoft Corporation.

```
{tp_PerUsersProperties}        image,
{tp_PartOrder}                 int,
{tp_Flags}                     int,
{AllUsers}                     int,
tp_Cache                       image,
{Per_tp_Cache}                 image,
{tp_ListId}                    nvarchar(38),
tp_Type                        tinyint,
tp_Source                      ntext,
tp_View                        ntext;
```

tp_Id: The Web Part Identifier (section 2.2.1.15) of the Web part. This value MUST NOT be NULL.

{tp_ZoneId}: The name of a Web part zone. This value can be NULL.

tp_WebPartTypeId: A 16-byte value uniquely identifying the type of the Web part. MUST NOT be NULL.

{tp_IsIncluded}: If 1, this indicates that the Web part is visible. This value MUST be 1 or 0.

{tp_FrameState}: A value that indicates the frame state of the Web part. FrameState is obsolete and is equivalent to ChromeState. This value MUST be one of the following.

Value	Description
0	Normal. The Web part is displayed in its normal state, with title, content, and placement within the page.
1	Minimized. The Web part is collapsed so that only the title portion of the frame appears.

tp_AllUsersProperties: Properties specified for all users. This value can be NULL.

{tp_PerUsersProperties}: Properties specified for per-user basis. This value can be NULL.

{tp_PartOrder}: Ordinal number that indicates the location of the Web part in relation to other Web parts in the same zone. This value can be NULL.

{tp_Flags}: A View Flags value that specifies view-related settings for this Web part. This value can be equal to 0.

{AllUsers}: A flag that indicates whether customization or personalization is in effect on this Web parts page. This value MUST be one of the following.

Value	Description
1	User not specified; changes apply to all users of this Web part instance. Same indicates customization mode.
2	Personalization is in effect; changes apply to the **tp_UserId** in the Personalization Table.
3	Personalization is in effect; changes apply to the **tp_UserId** in the Web Parts Table.

tp_Cache: Private data cache for the Web part. This value can be NULL.

{Per_tp_Cache}: Private data cache for documents at the published level; otherwise, this value MUST be NULL.

Copyright © 2014 Microsoft Corporation.

{tp_ListId}: The List Identifier (section 2.2.1.5) of the list (1) to which this Web part refers, enclosed in braces. If not referencing a list (1), this value MUST be NULL.

tp_Type: The Page Type of this Web part. This value can be NULL.

tp_Source: Properties of the Web part, as specified by a Windows SharePoint Services-compatible HTML editor. This value can be NULL.

tp_View: An ntext value that contains implementation-specific XML used when processing this Web part. If this Web part is not a view, this MUST be NULL.

3.1.5.15.17 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 view Web page-related metadata. It MUST return ONLY if the input **@PageView** is 0.

The Web Parts Metadata, Nonpersonalized Result Set 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.

```
tp_Id                       uniqueidentifier,
tp_ZoneID                   nvarchar(64),
tp_WebPartTypeId            uniqueidentifier,
tp_IsIncluded               bit,
tp_FrameState               tinyint,
tp_AllUsersProperties       image,
tp_PerUsersProperties       image,
tp_PartOrder                int,
{tp_Flags}                  int,
{AllUsers}                  int,
tp_Cache                    image,
{Per_tp_Cache}              image,
{tp_ListId}                 nvarchar(38),
tp_Type                     tinyint,
tp_Source                   ntext,
tp_View                     ntext;
```

tp_Id: The Web Part Identifier (section 2.2.1.15) of the Web part. This value MUST NOT be NULL.

tp_ZoneId: The name of a Web part zone. This value can be NULL.

tp_WebPartTypeId: A 16-byte value uniquely identifying the type of the Web part. MUST NOT be NULL.

tp_IsIncluded: If 1, this indicates that the Web part is visible. MUST NOT be NULL.

tp_FrameState: A value that indicates the frame state of the Web part. FrameState is obsolete and is equivalent to ChromeState. This value MUST be one of the following.

Value	Description
0	Normal. The Web part is displayed in its normal state, with title, content, and placement within

Copyright © 2014 Microsoft Corporation.

Value	Description
	the page.
1	Minimized. The Web part is collapsed so that only the title portion of the frame appears.

tp_AllUsersProperties: Properties specified for all users. This value can be NULL.

tp_PerUsersProperties: Properties specified for per-user basis. This value can be NULL.

tp_PartOrder: Ordinal number that indicates the location of the Web part in relation to other Web parts in the same zone. This value can be NULL.

{tp_Flags}: A View Flags value that specifies view-related settings for this Web part. This value MUST NOT be NULL.

{AllUsers}: A flag that indicates whether customization or personalization is in effect on this Web parts page. This value MUST be equal to 1, indicating customization.

tp_Cache: Private data cache for the Web part. This value can be NULL.

{Per_tp_Cache}: Private data cache for published documents. This value MUST be NULL.

{tp_ListId}: The List Identifier (section 2.2.1.5) of the list (1) to which this Web part refers, enclosed in braces. If not referencing a list (1), this value MUST be NULL.

tp_Type: The Page Type of this Web part. This value can be NULL.

tp_Source: Properties of the Web part as specified by a Windows SharePoint Services-compatible HTML editor. This value can be NULL.

tp_View: An ntext value containing implementation-specific XML used when processing this Web part. If this Web part is not a view, this MUST be NULL.

3.1.5.15.18 List Metadata Result Set

The List Metadata Result Set contains the metadata for the lists 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: Web Parts Metadata, Personalized (section 3.1.5.15.16) or Web Parts Metadata, Nonpersonalized, (section 3.1.5.15.17) as appropriate).

The List Metadata Result Set is part of a series of result sets describing view Web page-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 defined in the Common Result Sets List Metadata Result Set (section 2.2.5.12).

```
tp_Id                   uniqueidentifier,
tp_Title                nvarchar(255),
tp_Modified             datetime,
tp_Created              datetime,
tp_LastDeleted          datetime,
tp_Version              int,
```


Copyright © 2014 Microsoft Corporation.

```
tp_BaseType                    int,
tp_FeatureId                   uniqueidentifier,
tp_ServerTemplate              int,
DirName                        nvarchar(256),
LeafName                       nvarchar(128),
DirName                        nvarchar(256),
LeafName                       nvarchar(128),
tp_ReadSecurity                int,
tp_WriteSecurity               int,
tp_Description                 ntext,
{tp_Fields}                    ntext,
tp_Direction                   int,
{AnonymousPermMask}            bigint,
tp_Flags                       bigint,
tp_ThumbnailSize               int,
tp_WebImageWidth               int,
tp_WebImageHeight              int,
tp_ImageUrl                    nvarchar(255),
tp_ItemCount                   int,
tp_Author                      int,
tp_HasInternalFGP              bit,
tp_ScopeId                     uniqueidentifier,
{Acl}                          image,
tp_EventSinkAssembly           nvarchar(255),
tp_EventSinkClass              nvarchar(255),
tp_EventSinkData               nvarchar(255),
tp_EmailInsertsFolder          nvarchar(255),
tp_EmailInsertsLastSyncTime    nvarchar(50),
tp_EmailAlias                  nvarchar(128),
tp_WebFullUrl                  nvarchar(257),
tp_WebId                       uniqueidentifier,
tp_WebTitle                    nvarchar(255),
tp_WebTemplate                 int,
tp_WebLanguage                 int,
tp_WebCollation                smallint,
tp_SendToLocation              nvarchar(512),
{tp_MaxMajorVersionCount}      int,
{tp_MaxMajorwithMinorVersionCount}int,
tp_MaxRowOrdinal               int,
tp_ListDataDirty               int,
tp_DefaultWorkflowId           uniqueidentifier,
{tp_ContentTypes}              ntext,
tp_Subscribed                  bit;
```

tp_Id: The List Identifier (section 2.2.1.5) of the list (1).

tp_Title: The title of the list (1) for display in the user interface.

tp_Modified: A time stamp, in UTC format, specifying when the list (1) was last modified.

tp_Created: A time stamp, in UTC format, specifying when the list (1) was created.

tp_LastDeleted: A time stamp, in UTC format, specifying when an item was last deleted from the list (1).

tp_Version: A counter incremented any time a change is made to the schema or other properties of the list (1). Used for internal conflict detection.

Copyright © 2014 Microsoft Corporation.

tp_BaseType: Contains the List Base Type (section 2.2.3.11) value from which the list (1) is derived.

tp_FeatureId: The Feature Identifier (section 2.2.1.4) for the feature that defines the base schema of the list (1).

tp_ServerTemplate: Contains the List Server Template (section 2.2.3.12) enumeration value of the list template that defines the base structure of the list (1).

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

LeafName: A leaf name of the location that contains the list (1).

DirName: The directory name of the default template of the list (1). This value can be NULL if a template document is not defined for the list (1).

LeafName: The leaf name of the default template of the list (1). This value can be NULL if a template document is not defined for the list (1).

tp_ReadSecurity: Special restrictions that can be placed on list item access. The following values are valid.

Value	Description
1	No special restrictions.
2	Users will see only their own list items. The WFE MUST NOT display list items to users without the ManageLists right unless the list item was created by that user (for example, tp_Author = @UserId).

tp_WriteSecurity: Special restrictions that can be placed on list item update. The following values are valid.

Value	Description
1	No special restrictions.
2	Users will see only their own list items. The WFE MUST NOT allow users without the ManageLists right to update a list item unless the list item was created by that user (for example, tp_Author = @UserId).
4	Users will not update any list items in the list (1). The WFE MUST NOT allow users without the ManageLists right to add or update list items in the list (1).

tp_Description: The description of the list (1) for display in the user interface.

{tp_Fields}: Contains a WSS implementation-specific version string followed by an XML fragment representation of the field (2) definitions of the list (1), as specified in the **FieldDefinitionDatabaseWithVersion** type defined in section 2.2.8.3.5.

tp_Direction: An enumerated value specifying the direction of text flow for user interface elements presented by the list (1). The following values are valid.

Value	Description
0	No explicit direction is specified.

Copyright © 2014 Microsoft Corporation.

Value	Description
1	Text flow is left to right.
2	Text flow is right to left.

{AnonymousPermMask}: A WSS Rights Mask that indicates the rights granted to a user that is anonymous, or has no specific rights, on the list (1). This field (2) MUST be NULL if the scope of **@SiteId** is equal to the scope of **@ListId**.

tp_Flags: A List Flags (section 2.2.2.5) value describing the list (1).

tp_ThumbnailSize: tp_ThumbnailSize: The width, in pixels, that SHOULD be used when creating thumbnail images of list items within this list (1). Thumbnails and Web images are WFE-generated images of documents that are implementation-specific capabilities of Windows SharePoint Services.

tp_WebImageWidth: The width, in pixels, that SHOULD be used when creating Web images of list items within this list (1).

tp_WebImageHeight: The height, in pixels, that SHOULD be used when creating Web images of list items within this list (1).

tp_ImageUrl: The URL of the image used to represent the list (1).

tp_ItemCount: The number of list items that are stored within the list (1).

tp_Author: The identifier of the user who is listed as creating the list (1).

tp_HasInternalFGP: This flag is set to 1 if there have ever been list items for the list (1) that have had a unique ACL applied. Otherwise, this flag is set to 0.

tp_ScopeId: Unique identifier of the security scope for the list (1). This indicates the specific WSS ACL Format (section 2.2.4.6) ACL to use for calculating the permission settings on the list (1).

{Acl}: The binary serialization of the WSS ACL Format ACL for the list (1). This MAY be used for the lists permissions, depending on the **tp_ScopeId** value of the list (1).

tp_EventSinkAssembly: The name of the assembly that contains the class definition of the event sink associated with the list (1).

tp_EventSinkClass: The name of the class definition for the event sink associated with the list (1).

tp_EventSinkData: This field (2) contains Unicode string data specific to the implementation of the event sink associated with the list (1).

tp_EmailInsertsFolder: This field contains a URL fragment specifying the directory on the configured email inserts server that SHOULD be inspected for new email messages to be processed for the list (1). If the List Flags for the list (1) do not have the value 0x0000000000010000 set, this parameter MUST be ignored.

tp_EmailInsertsLastSyncTime: This field (2) is a UTC time stamp encoded as a Unicode string in yyyy-mm-dd hh**:**mi**:**ss.mmm format specifying the last time the location specified in the **tp_EmailInsertsFolder** column was inspected for new list items. If the List Flags value for the list (1) does not have the value 0x0000000000010000 set, this parameter MUST be ignored.

tp_EmailAlias: The email alias of the list (1). This alias is used to allow files to be sent directly to the list (1) through an implementation-specific email handling feature.

Copyright © 2014 Microsoft Corporation.

tp_WebFullUrl: The complete store-relative form URL to the site (2) that contains the list (1).

tp_WebId: The Site Identifier (section 2.2.1.11) of the site (2) that contains the list (1).

tp_WebTitle: The title used for display in the user interface of the site (2) that contains the list (1).

tp_WebTemplate: The identifier of the site template for the site (2) that contains the list (1).

tp_WebLanguage: The LCID of the display language of the site (2) that contains the list (1).

tp_WebCollation: The collation order for information in the site (2) that contains the list (1).

tp_SendToLocation: The title and URL for the "Send To Location" configured on the list (1). The **SendToLocation** is an implementation-specific feature that allows users to manually save copies of list items and documents to the remote location.

{tp_MaxMajorVersionCount}: If the list has versioning enabled, this field (2) contains the number of major versions that will be retained for this document. All versions more than **tp_MaxMajorVersionCount** removed from the current version of the document are automatically removed at version creation time. A value of 0 specifies that versions SHOULD NOT automatically be removed for the list (1).

{tp_MaxMajorwithMinorVersionCount}: If the list (1) has versioning enabled, this field (2) contains the number of major versions that will have their associated minor versions retained for this document. All versions more than **tp_MaxMajorwithMinorVersionCount** removed from the current version of the document are automatically removed at version creation time. A value of 0 specifies that versions SHOULD NOT automatically be removed for the list (1).

tp_MaxRowOrdinal: This specifies the maximum row ordinal used to store list items for the list (1). This value indicates an implementation-specific calculation for storage of list items within lists.

tp_ListDataDirty: This is set to 1 if the list items in the list (1) require dependency update processing before their next access (for example, updating document link information by parsing each document).

tp_DefaultWorkflowId: The Workflow Identifier (section 2.2.1.16) corresponding to the workflow (2) to be invoked if the document is in a moderated list (1) and the document is submitted for approval as part of a check in. If the document does not exist or is not contained in a list (1) with a configured approval workflow (2), this value MUST be NULL.

{tp_ContentTypes}: Implementation-specific XML data specifying the content types registered for the list (1).

tp_Subscribed: Set to 1 if an alert for changes to the list (1) has been created in the past, signifying that additional processing needs to be performed.

3.1.5.15.19 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 List Metadata Result Set (section 2.2.5.12) 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 defined in the Common Result Sets Event Receivers Result Set (section 2.2.5.9).

Copyright © 2014 Microsoft Corporation.

3.1.5.15.20 List Security Information Result Set

The List Security Information Result Set contains permissions information for the lists associated with the Web parts that appear on the specified document.

The List Security Information Result Set MUST return only if requested (input parameter **@PrefetchListScope** is set to 1) and if such lists exist (the List Metadata Result Set (section 2.2.5.12) returns with at least one row).

The List Security Information Result Set returns one row per each unique scope associated with the lists.

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

ListId: The List Identifier (section 2.2.1.5) of the list (1).

ScopeId: The Scope Identifier (section 2.2.1.8) of the Permissions Scope that applies to the list (1). This MUST be set to zero if the list (1) does not have FGP.

Acl: Contains the ACL of the Permissions Scope that applies to the list (1). This MUST be NULL if ScopeId is zero.

AnonymousPermMask: Contains the WSS Rights Mask (section 2.2.2.13) on the list (1) in effect for anonymous users. This MUST be 0 if **ScopeId** is zero.

3.1.5.15.21 List Web Parts Result Set

The List Web Parts Result Set contains information about the Web parts related to the lists associated with the specified document.

The List Web Parts Result Set MUST return only if such lists exist (the List Metadata Result Set (section 2.2.5.12) 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 defined in the Common Result Sets List Web Parts Result Set (section 2.2.5.13).

3.1.5.15.22 Content Type Order Result Set

The Content Type Order Result Set provides the information necessary for the implementation-specific rendering of a list View web page. The information necessary to render the content types in the configured order, if such exists, is provided in a binary image value.

The 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.

```
{CurrentFolderURL}      varchar,
{MetaInfo}              image
```

{CurrentFolderURL}: The URL of the current folder as specified in the **@CurrentFolderURL** input parameter.

Copyright © 2014 Microsoft Corporation.

{MetaInfo}: The information about content type order, embedded in a binary image. If no information is available for the current folder, or if there is no current folder specified, or if the document is not contained in a list (1), then this parameter MUST be NULL and the column is unnamed. Otherwise, the column MUST be named 'MetaInfo'.

3.1.5.15.23 Current Folder Scope Result Set

The Current Folder Scope Result set contains scope information about the current folder as specified by the input parameter **@CurrentFolderURL**. It is used by the WFE to security trim the information rendered to the user in a view webpage based on the user's access.

The Current Folder Scope Result Set MUST return only if **@PageView** is not NULL, and it MUST NOT return if **proc_FetchDocForHttpGet** returns a code of 1271, indicating that the site collection was locked. In addition, the Current folder Scope Result Set MUST return only if the document is contained in a list (1) and resides in a folder that is not the list's root folder.

If the Current Folder Scope Result Set returns, it MUST return one row.

```
{FolderScopeId}              uniqueidentifier,
{FolderId}                   int;
```

{FolderScopeId}: The Scope Identifier (section 2.2.1.8) of the security scope effective for the specified folder.

{FolderId}: The integer identifier of the folder document within the containing list (1).

3.1.5.15.24 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 (2) 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 the Common Result Sets Site Feature List Result Set (section 2.2.5.21).

3.1.5.15.25 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 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 site's navigation hierarchy, excluding the site's own scope.

The Navigation Context Security Information Result Set is defined in the Common Result Sets Unique Permissions Result Set (section 2.2.5.25).

Copyright © 2014 Microsoft Corporation.

3.1.5.15.26 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 return only upon successful execution if **@PageView** is not NULL and the security information about the site (2) or the parent site in the site's navigation hierarchy was larger than 1,800 bytes.

If the NULL Navigation Context Security Information Result Set returns, it MUST return one row, as defined in the Common Result Sets NULL Unique Permissions Result Set (section 2.2.5.15).

3.1.5.15.27 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 (section 3.1.5.15.25) nor the NULL Navigation Context Security Information (section 3.1.5.15.26) Result Sets return.

If the Empty Navigation Context Security Information Result Set returns, it MUST return zero rows, as defined using T-SQL syntax in the Common Result Sets Empty Result Set (section 2.2.5.8) section.

3.1.5.16 proc_FetchDocForRead

The **proc_FetchDocForRead** stored procedure is invoked to request the metadata information and content of a document.

```
PROCEDURE proc_FetchDocForRead(
      @DocSiteId                    uniqueidentifier,
      @DocWebId                     uniqueidentifier,
      @DocDirName                   nvarchar(256),
      @DocLeafName                  nvarchar(128),
      @DocFullUrl                   nvarchar(260),
      @LooksLikeAttachmentFile      bit,
      @GetContent                   bit,
      @GetWebListForNormalization   bit,
      @bGetContainingList           bit,
      @bCheckout                    bit,
      @UserId                       int,
      @Version                      int,
      @ChunkSize                    int,
      @MaxLevel                     tinyint,
      @Level                        tinyint OUTPUT
);
```

@DocSiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the requested document.

@DocWebId: The Site Identifier (section 2.2.1.11) of the site (2) containing the requested document.

@DocDirName: The directory name of the requested document.

Copyright © 2014 Microsoft Corporation.

@DocLeafName: The leaf name of the requested document.

@DocFullUrl: This parameter is unused.

@LooksLikeAttachmentFile: Specifies whether the requested document appears to the WFE to be an attachment to a list item. If this flag is set to "1", **proc_FetchDocForRead** MUST determine whether the current user has sufficient permissions to retrieve the requested document as an attachment.

@GetContent: A bit flag specifying whether to return the content of the document or not. If this flag is set to "1", **proc_FetchDocForRead** MUST return the content of the requested document in the Document Information and Content Result Set (section 3.1.5.16.10) or the Document Version Information and Content Result Set (section 3.1.5.16.11).

@GetWebListForNormalization: A bit flag specifying whether to return the subsite of the site (2) specified by **@DocWebId**. If this flag is set to "1", **proc_FetchDocForRead** MUST return a list of child sites in the Subsite List Result Set (section 3.1.5.16.1).

@bGetContainingList: Specifies whether to return the event receivers information about the list (1) containing the requested document. If this flag is set to "1", **proc_FetchDocForRead** MUST return the Event Receivers Result Set (section 3.1.5.16.6).

@bCheckout: Specifies whether the current user is requesting to check out the document. If this flag is set to "1", and the **@LooksLikeAttachmentFile** parameter is set to "1", **proc_FetchDocForRead** MUST determine whether the current user has sufficient permissions to check out the requested document as an attachment.

@UserId: The User Identifier (section 2.2.1.13) for the current user requesting the information. This is used by **proc_FetchDocForRead** for permission checking.

@Version: Specifies the UI version number of the document that is being requested. A value of "-1" specifies the most recent version of the document.

@ChunkSize: Specifies the maximum size, in bytes, of the document content image to be returned in the Document Information and Content Result Set. If the document content image size is larger than **@ChunkSize**, a single zero byte is returned as the document content image, and the WFE can request the content of the document in a subsequent operation.

@MaxLevel: A Publishing Level Type (section 2.2.2.6) value that indicates the maximum publishing level of the document to be returned in **@Level** if multiple levels of the document are available and the current user is not the owner of the draft or does not have the document checked out.

@Level: The Publishing Level Type value of the requested version of the document visible to the current user, returned as an output parameter. This value MUST be returned as NULL if the document does not exist.

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

Value	Description
0	Successful execution.
3	The document does not exist.

The **proc_FetchDocForRead** stored procedure returns a Publishing Level Type value in the parameters as described earlier. **proc_FetchDocForRead** MUST return multiple result sets. Some

Copyright © 2014 Microsoft Corporation.

of the result sets MUST NOT be returned depending upon input parameters or calculations performed in **proc_FetchDocForRead**, and all result sets that are returned MUST be sent in the order defined as follows.

3.1.5.16.1 Subsite List Result Set

The Subsite List Result Set contains an unordered list of store-relative form URLs for all subsites within the site collection whose parent site (2) is specified in the **@DocWebId** parameter.

The Subsite List Result Set MUST only be returned if **@GetWebListForNormalization** is set to "1". The Subsite List Result Set MUST contain one row for each subsite with the specified parent site (2), and it MUST contain no rows if there are no such subsites.

The Subsite List Result Set is defined in the URL Result Set (section 2.2.5.26).

3.1.5.16.2 Link Info Single Doc Result Set

The Link Info Single Doc Result Set contains information about links to or within the requested document. Entries are present both for forward links and for backward links. The Link Info Single Doc Result Set MUST be returned, and it MUST contain one row for each link that is referenced.

The Link Info Single Doc Result Set is defined in the Common Result Sets Link Information Result Set (section 2.2.5.11).

3.1.5.16.3 Document Metadata Result Set

The Document Metadata Result Set contains the metadata about the most current version of the requested document visible to the current user.

The Document Metadata Result Set MUST only be returned if **@**Version is negative. There MUST be one row in the Document Metadata Result Set if the document exists; otherwise, there MUST be no rows.

The Document Metadata Result Set is defined in the Common Result Sets Document Metadata Result Set (section 2.2.5.6).

3.1.5.16.4 Document Version Metadata Result Set

The Document Version Metadata Result Set contains the metadata about the requested version of the document.

The Document Version Metadata Result Set MUST only be returned if **@**Version is not negative. If the document exists, the Document Version Metadata Result Set MUST contain one row; otherwise, it MUST contain zero rows.

The Document Version Metadata Result Set is defined in the Common Result Sets Document Version Metadata Result Set (section 2.2.5.6)

3.1.5.16.5 NULL Result Set

The NULL Result Set is a placeholder that returns no data.

The NULL Result Set MUST only be returned if **@**Version is negative, the requested document exists, and **@DocWebId** is NULL. The NULL Result Set MUST contain zero rows in a schema containing a single unnamed column.

Copyright © 2014 Microsoft Corporation.

3.1.5.16.6 Event Receivers Result Set (1)

The Event Receivers Result Set contains information about the event receivers defined for this document.

The Event Receivers Result Set MUST only be returned if one of these conditions is true:

- The requested document exists and either **@Version** is not negative, OR
- **@Version** is negative and **@DocWebId** is not NULL.

The Event Receivers Result Set MUST contain one row for each event receiver registered with an Event Host Type (section 2.2.3.5) of 3 (list item) for this document.

The Event Receivers Result Set is defined in the Common Result Sets Event Receivers Result Set (section 2.2.5.9).

3.1.5.16.7 List Metadata Result Set

The List Metadata Result Set contains the metadata associated with the list (1) that contains the requested document.

The List Metadata Result Set MUST only be returned if the **@bGetContainingList** parameter is set to "1" and the document is contained within a list (1). The List Metadata Result Set MUST contain one row.

The List Metadata Result Set is defined in the Common Result Sets List Metadata Result Set (section 2.2.5.12).

3.1.5.16.8 Empty List Result Set

The Empty List Result Set contains a single, unnamed column with no rows to indicate that the document is not contained within a list (1). The Empty List Result Set MUST only be returned if the @bGetContainingList parameter is set to "1" and the document is not contained within a list (1).

The Empty List Result Set is defined in the Common Result Sets Empty Result Set (section 2.2.5.8).

3.1.5.16.9 Event Receivers Result Set (2)

The Event Receivers Result Set contains the event receivers associated with the list (1) that contains the requested document.

The Event Receivers Result Set MUST only be returned if the **@bGetContainingList** parameter is set to "1" and the document is contained within a list (1). The Event Receivers Result Set MUST contain one row for each event receiver registered with an Event Host Type (section 2.2.3.5) of 2 (List) for the List.

The Event Receivers Result Set is defined in the Common Result Sets Event Receivers Result Set (section 2.2.5.9).

3.1.5.16.10 Document Information and Content (Read) Result Set

The Document Information and Content (Read) Result Set contains information about the content of the document stream for the requested document.

Copyright © 2014 Microsoft Corporation.

The Document Information and Content (Read) Result Set MUST only be returned if **@GetContent** is set to "1" and **@**Version is negative. If the document exists, the Document Information and Content (Read) Result Set MUST contain one row; otherwise, it MUST contain zero rows.

The Document Information and Content (Read) Result Set is similar to the Document Information and Content (Update) Result Set, except for column naming.

```
{Size}                  int,
SetupPathVersion        tinyint,
SetupPath               nvarchar(255),
SetupPathUser           nvarchar(255),
Content                 image,
Dirty                   bit,
DocFlags                int,
DoclibRowId             int,
VirusVendorID           int,
VirusStatus             int,
VirusInfo               nvarchar(255),
Version                 int,
Id                      uniqueidentifier,
Version                 int;
```

{Size}: The size of the requested document in bytes.

SetupPathVersion: For a ghosted document, this governs the setup path location to which the **SetupPath** fragment is relative. This value MUST be NULL if the document does not exist and is undefined for a document that was never ghosted. The following values are valid.

Value	Description
2	The SetupPath is relative to the install location of Windows SharePoint Services 2.0 on the WFE (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\60).
3	The SetupPath is relative to the install location of Windows SharePoint Services 3.0 on the WFE (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\12).

SetupPath: For a document that is now or once was ghosted, **SetupPath** MUST contain the setup path fragment relative to the base setup path described by the **SetupPathVersion** value, where the content stream of this document can be found. This value MUST be NULL in the case of a document that does not exist or a document that was never ghosted.

SetupPathUser: If the document is now or once was ghosted, **SetupPathUser** MUST contain the login name of the user who created the ghosted document. This value MUST be NULL in the case of a document that does not exist, and it is undefined for a document that was never ghosted.

Content: The document stream content of the document. For a ghosted document, content MUST be NULL. If the content is larger than the value specified in the **@ChunkSize** parameter, a single zero byte MUST be returned, and the WFE can request individual chunks of content in a subsequent request.

Dirty: A bit set to "1" to indicate that this document MUST have dependency update processing performed before its document stream is returned to an external agent. This field's value is implementation-dependent if this document does not have a document stream and MUST be ignored.

Copyright © 2014 Microsoft Corporation.

DocFlags: A Doc Flags (section 2.2.2.3) value describing the document. This value MUST be NULL if the document does not exist.

DocLibRowId: The identifier of the row in the document library that represents this document. This value MUST be 0 if the document is not stored in a list (1).

VirusVendorID: The identifier of the virus scanner that processed this document. This value MUST be NULL if the document does not exist or if this document has not been processed by a virus scanner.

VirusStatus: A Virus Status (section 2.2.3.18) value specifying the current virus scan status of this document. This value MUST be NULL if the document does not exist or if this document has not been processed by a virus scanner.

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 the document does not exist or if the document has not been processed by a virus scanner.

Version: 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 internal version counter value for the document in the Docs View (section 2.2.7.4), except in the Document Version Information and Content Result Set (section 3.1.5.16.11), where this value MUST be NULL.

Id: The document identifier (section 2.2.1.2) of this document.

Version: An internal version counter incremented any time a change is made to this document and used for internal conflict detection. This value MUST be equal to the previous version column, except in the Document Version Information and Content Result Set, where this value MUST be set to the value of the **@Version** parameter.

3.1.5.16.11 Document Version Information and Content Result Set

The Document Version Information and Content Result Set contains information about the content of the document stream for the requested document.

The Document Version Information and Content Result Set MUST only be returned if **@GetContent** is set to "1" and **@Version** is not negative.

The Document Version Information and Content Result Set MUST contain one row if the specified document exists; otherwise, it MUST contain no rows.

```
{Size}                  int,
{SetupPathVersion}      tinyint,
{SetupPath}             nvarchar(255),
{SetupPathUser}         nvarchar(255),
{Content}               image,
{Dirty}                 bit,
DocFlags                int,
{DoclibRowId}           int,
VirusVendorID           int,
VirusStatus             int,
VirusInfo               nvarchar(255),
{Version}               int,
Id                      uniqueidentifier,
{Version}               int;
```


Copyright © 2014 Microsoft Corporation.

For the field (2) descriptions, see the Document Information and Content (Read) Result Set (section 3.1.5.16.10) section.

3.1.5.16.12 Attachment State Result Set

The Attachment State Result Set contains the information about the attachment state of the requested document. The Attachment State Result Set MUST be returned and MUST contain one row.

The Attachment State Result Set is defined in the Common Result Sets Attachment State Result Set (section 2.2.5.2).

3.1.5.16.13 Audit Mask Result Set

The Audit Mask Result Set contains the information about the Audit Flags (section 2.2.2.1) associated with this document. The Audit Mask Result Set MUST be returned and MUST contain a single row of data if the document exists.

The Audit Mask Result Set is defined in the Common Result Sets Site Audit Mask Result Set (section 2.2.5.20).

3.1.5.17 proc_FetchDocForUpdate

The **proc_FetchDocForUpdate** stored procedure is invoked to request document content and metadata information. It also updates the **CacheParseId** flag for the requested document if the **@CacheParse** parameter is set to "1".

```
PROCEDURE proc_FetchDocForUpdate(
      @DocSiteId                    uniqueidentifier,
      @DocWebId                     uniqueidentifier,
      @DocDirName                   nvarchar(256),
      @DocLeafName                  nvarchar(128),
      @UserId                       int,
      @Version                      int,
      @GetContent                   bit,
      @GetWebListForNormalization   bit,
      @CacheParse                   bit,
      @GetWebPartInfo               bit,
      @bGetContainingList           bit,
      @LooksLikeAttachmentFile      bit,
      @ChunkSize                    int,
      @Level                        tinyint           OUTPUT
);
```

@DocSiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the requested document.

@DocWebId: The Site Identifier (section 2.2.1.11) of the site (2) containing the requested document.

@DocDirName: The directory name of the requested document.

@DocLeafName: The leaf name of the requested document.

@UserId: The User Identifier (section 2.2.1.13) for the current user requesting the information. This is used by **proc_FetchDocForUpdate** for permission checking.

Copyright © 2014 Microsoft Corporation.

@Version: Specifies the UI version number of the document being requested. A value of "-1" specifies the most recent version of the document.

@GetContent: A bit flag specifying whether or not to return the content of the document. This flag MUST be set to "1" for the server to return either the Document Information and Content (Update) Result Set(section 3.1.5.17.13), the Document Version 1 Information and Content Result Set (section 3.1.5.17.14), or the Document Version 2 Information and Content Result Set (section 3.1.5.17.15).

@GetWebListForNormalization: A bit flag specifying whether to return the subsite of the site (2) specified by **@DocWebId**. If this flag is set to "1", **proc_FetchDocForUpdate** MUST return a list of child sites in the Subsite List Result Set (section 3.1.5.17.1).

@CacheParse: A bit flag specifying whether to update the **CacheParseId** flag of the requested document. If this flag is set to "1", **proc_FetchDocForUpdate** MUST update the **CacheParseId** flag in the Docs View (section 2.2.7.4) for the requested document.

@GetWebPartInfo: A bit flag specifying whether to return the Web parts information for the requested document. If this flag is set to "1", **proc_FetchDocForUpdate** MUST return the Web parts information for the requested document in the Web Part Info Result Set (section 3.1.5.17.8) and the Zone ID Result Set (section 3.1.5.17.9).

@bGetContainingList: Specifies whether to return the event receivers information about the list containing the requested document. If this flag is set to "1", **proc_FetchDocForUpdate** MUST return the Event Receivers Result Set (section 2.2.5.9) for the list.

@LooksLikeAttachmentFile: Specifies whether the requested document appears to the WFE to be an attachment to a list item. If this flag is set to "1", **proc_FetchDocForUpdate** MUST determine whether the current user has sufficient permissions to retrieve the requested document as an attachment.

@ChunkSize: Specifies the maximum size, in bytes, of the document content image to be returned in the Document Information and Content (Update) Result Set. If the document content image size is larger than **@ChunkSize**, a single zero byte is returned as the document content image, and the WFE can request the content of the document in a subsequent operation.

@Level: The Publishing Level Type (section 2.2.2.6) value of the requested version of the document visible to the current user, returned as an output parameter. This value MUST be returned as NULL if the document does not exist.

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

The **proc_FetchDocForUpdate** stored procedure returns a Publishing Level Type value in the parameters as specified earlier. **proc_FetchDocForUpdate** MUST return multiple result sets. Some of the result sets MUST NOT be returned, depending upon input parameters or calculations performed in **proc_FetchDocForUpdate**, and all result sets that are returned MUST be sent in the order defined as follows.

3.1.5.17.1 Subsite List Result Set

The Subsite List Result Set contains an unordered list of store-relative form URLs for all subsites within the site collection whose parent Site is specified in the **@DocWebId** parameter.

The Subsite List Result Set MUST only be returned if **@GetWebListForNormalization** is set to "1." The Subsite List Result Set MUST contain one row for each subsite within the specified parent Site, and MUST contain no rows if there are no such subsites.

Copyright © 2014 Microsoft Corporation.

The Subsite List Result Set is defined in the URL Result Set (section 2.2.5.26).

3.1.5.17.2 ACL and Permission Result Set

The ACL and Permission Result Set contains information about the permissions associated with the scope in effect for the document. The ACL and Permission Result Set MUST be returned and MUST contain one row. If the document does not exist, the values of both columns MUST be NULL. The ACL and Permission Result Set is defined in the Common Result Sets ACL and Permission Result Set (section 2.2.5.1).

3.1.5.17.3 Document Metadata Result Set

The Document Metadata Result Set contains the metadata about the most current version of the requested document visible to the current user.

The Document Metadata Result Set MUST only be returned if **@**Version is negative. If the document exists, the Document Metadata Result Set MUST contain one row; otherwise, it MUST contain no rows.

The Document Metadata Result Set is defined in the in the Common Result Sets Document Metadata Result Set (section 2.2.5.6).

3.1.5.17.4 Document Version Metadata Result Set

The Document Version Metadata Result Set contains the metadata about the requested version of the document.

The Document Version Metadata Result Set MUST only be returned if **@**Version is not negative. If the document exists, the Document Version Metadata Result Set MUST contain one row; otherwise, it MUST contain no rows.

The Document Version Metadata Result Set is defined in the Common Result Sets Document Version Metadata Result Set (section 2.2.5.7).

3.1.5.17.5 NULL Result Set

The NULL Result Set is a placeholder that returns no data.

The NULL Result Set MUST only be returned if **@**Version is negative, the requested document exists, and **@**DocWebId is NULL. The NULL Result Set MUST contain zero rows in a schema containing a single unnamed column.

3.1.5.17.6 Event Receivers Result Set (1)

The Event Receivers Result Set contains information about the event receivers defined for this document.

The Event Receivers Result Set MUST only be returned if the requested document exists in the site (2) specified by **@DocWebId**. The Event Receivers Result Set MUST contain one row for each event receiver registered with an Event Host Type (section 2.2.3.5) of 3 (list item) for this document.

The result set is defined in the Common Result Sets Event Receivers Result Set (section 2.2.5.9).

Copyright © 2014 Microsoft Corporation.

3.1.5.17.7 Link Info Single Doc Fixup Result Set

The Link Info Single Doc Fixup Result Set contains information about forward links within the requested document. The Link Info Single Doc Fixup Result Set MUST only be returned if Link information was requested by setting **@CacheParse** to "1."

```
LinkDirName                 nvarchar(256),
LinkLeafName                nvarchar(128),
LinkType                    tinyint,
LinkSecurity                tinyint,
LinkDynamic                 tinyint,
LinkServerRel               bit,
LinkStatus,                 tinyint,
PointsToDir                 bit,
WebPartId                   uniqueidentifier,
LinkNumber                  int,
WebId                       uniqueidentifier,
Search                      ntext,
FieldId                     uniqueidentifier;
```

LinkDirName: Contains the directory name of the link.

LinkLeafName: Contains the leaf name of the link.

LinkType: The LinkType (section 2.2.3.10) value of the link in the specified document.

LinkSecurity: A LinkSecurity (section 2.2.3.9) value specifying whether the scheme of the link is HTTP or HTTPS.

LinkDynamic: A LinkDynamic (section 2.2.3.8) value that specifies whether this link is one of several special link types.

LinkServerRel: Indicates whether the Link URL is server-relative (1) or not (0).

LinkStatus: A Document Store Type (section 2.2.2.4) indicating the type of document the Link points to. If the Link is a forward link for a document that doesn't exist, this MUST be NULL. This value MUST be NULL if this Link entry refers to a location that exists outside the site collection where the document is stored, or if it refers to a location that could not be verified. See Document Store Type for a list of valid values.

PointsToDir: If the Link pointed to a directory where a welcome page existed (for example, pointing to http**:**//server when a welcome page like http://server/default.aspx existed), the Link is automatically changed to be the URL to the welcome page itself. This bit MUST be set to true if this operation has been performed, so that it can be distinguished from an explicit Link to the welcome page.

WebPartId: If this Link corresponds to a web part within a web part zone, which is therefore logically part of this document, but not physically present in the HTML source code of this document, this is the GUID of the web part to which the Link belongs.

LinkNumber: An ordinal value denoting the relative order of the Link within the source of the document, web part, or field (2) being processed.

WebId: MUST be NULL.

Search: MUST be NULL.

Copyright © 2014 Microsoft Corporation.

FieldId: If the Link is for a list item field (2) within this document, this is the GUID of the field to which the Link belongs.

3.1.5.17.8 Web Part Info Result Set

The Web Part Info Result Set MUST only be returned when **@GetWebPartInfo** is set to "1." If web part data for the requested document exists, one row MUST be returned for each web part, otherwise zero rows MUST be returned.

```
tp_Id                       uniqueidentifier,
tp_Level                    tinyint
tp_Source                   ntext,
tp_AllUsersProperties       image;
```

tp_Id: The Web Part Identifier (section 2.2.1.15) of the web part.

tp_Level: Publishing Level Type (section 2.2.2.6) value of the document.

tp_Source: Properties of the web part.

tp_AllUsersProperties: A list of the XML properties which are common for all users of the web part.

3.1.5.17.9 Zone ID Result Set

The Zone ID Result Set returns a list of web part zone names. The Zone ID Result Set MUST only be returned when **@GetWebPartInfo** is set to "1". If web part data for the requested document exists, one row MUST be returned for each web part zone; otherwise zero rows MUST be returned.

```
tp_ZoneID                   nvarchar(64);
```

tp_ZoneID: The name of a web part zone.

3.1.5.17.10 List Metadata Result Set

The List Metadata Result Set contains the metadata associated with the List containing the requested document.

The List Metadata Result Set MUST only be returned if the **@bGetContainingList** parameter is set to "1" and the document is contained within a List. The List Metadata Result Set MUST contain one row.

The List Metadata Result Set is defined in the Common Result Sets List Metadata Result Set (section 2.2.5.12).

3.1.5.17.11 Empty List Result Set

The Empty List Result Set contains a single unnamed column with no rows to indicate the document is not contained within a List. The Empty List Result Set MUST only be returned if the **@bGetContainingList** parameter is set to "1" and the document is not contained within a List.

The Empty List Result Set is defined in the Common Result Sets Empty Result Set (section 2.2.5.8).

Copyright © 2014 Microsoft Corporation.

3.1.5.17.12 Event Receivers Result Set (2)

The Event Receivers Result Set contains the event receivers associated with the List which contains the requested document.

The Event Receivers Result Set MUST only be returned if the **@bGetContainingList** parameter is set to "1" and the document is contained within a List. There MUST be one row in the Event Receivers Result Set for each event receiver registered with an Event Host Type (section 2.2.3.5) of 2 (List) for the List.

The Event Receivers Result Set is defined in the Common Result Sets Event Receivers Result Set (section 2.2.5.9).

3.1.5.17.13 Document Information and Content (Update) Result Set

The Document Information and Content (Update) Result Set contains information about the content of the document stream for the current version of the requested document.

The Document Information and Content (Update) Result Set MUST only be returned if **@GetContent** is set to "1" and **@**Version is negative. If the publishing level of the document specified by **@Level** exists, the Document Information and Content (Update) Result Set MUST contain one row; otherwise, the Document Information and Content (Update) Result Set MUST contain no rows.

The Document Information and Content (Update) Result Set is similar to the Document Information and Content (Read) Result Set, except for a number of columns that are unnamed here.

```
{Size}                  int,
SetupPathVersion        tinyint,
SetupPath               nvarchar(255),
SetupPathUser           nvarchar(255),
{Content}               image,
Dirty                   bit,
DocFlags                int,
{DoclibRowId}           int,
{VirusVendorID}         int,
{VirusStatus}           int,
{VirusInfo}             nvarchar(255),
Version                 int,
Id                      uniqueidentifier,
Version                 int;
```

{Size}: The size of the requested document in bytes.

SetupPathVersion: For a ghosted document, this governs the setup path location which the SetupPath fragment is relative to. This value MUST be NULL if the document does not exist, and is undefined for a document which was never ghosted. The following are valid values:

Value	Description
2	The SetupPath is relative to the install location of Windows SharePoint Services 2.0 on the WFE (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\60)
3	The SetupPath is relative to the install location of Windows SharePoint Services 3.0 on the WFE (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\12).

Copyright © 2014 Microsoft Corporation.

SetupPath: For a document which is now or once was ghosted, this parameter MUST contain the setup path fragment relative to the base setup path described by the **SetupPathVersion** value, where the content stream of this document can be found. In the case of a document which does not exist or a document which was never ghosted, this value MUST be NULL.

SetupPathUser: If the document is now or once was ghosted, this parameter MUST contain the login name of the User who created the ghosted document. In the case of a document which does not exist, this value MUST be NULL, and is undefined for a document which was never ghosted.

{Content}: The document stream content of the document. For a ghosted document, this MUST be NULL. If the content is larger than the value specified in the **@**ChunkSize parameter, a single zero byte MUST be returned, and the WFE can request individual chunks of content in a subsequent request.

Dirty: A bit set to "1," indicating that this document MUST have dependency update processing performed before its document stream is returned to an external agent. This field's value is implementation-dependent if this document does not have a document stream, and MUST be ignored.

DocFlags: A Doc Flags (section 2.2.2.3) value describing the document. This value MUST be NULL if the document does not exist.

{DocLibRowId}: The identifier of the row in the document library which represents this document. This value MUST be NULL.

{VirusVendorID}: The identifier of the virus scanner which processed this document. This value MUST be NULL.

{VirusStatus}: A Virus Status (section 2.2.3.18) value specifying the current virus scan status of this document. This value MUST be NULL.

{VirusInfo}: A string containing a provider specific message returned by the virus scanner when it last processed the document. This value MUST be NULL.

Version: An internal version counter incremented any time a change is made to this document, and used for internal conflict detection. This value MUST be set to the current internal version counter value for the document in the Docs View (section 2.2.7.4), except in the Document Version 1 Information and Content Result Set (section 3.1.5.17.14) and the Document Version 2 Information and Content Result Set (section 3.1.5.17.15), where this value MUST be NULL.

Id: The document identifier (section 2.2.1.2) of this document.

Version: An internal version counter incremented any time a change is made to this document, used for internal conflict detection. This value MUST be equal to the previous document column, except in the Document Version 1 Information and Content Result Set and the Document Version 2 Information and Content Result Set, where this value MUST be set to the value of the **@**Version parameter.

3.1.5.17.14 Document Version 1 Information and Content Result Set

The Document Version 1 Information and Content Result Set contains information about the content of the document stream for the requested version of the document.

The Document Version 1 Information and Content Result Set MUST only be returned if **@GetContent** is set to "1" and **@Version** is not negative, and the document exists with the publishing level specified by **@Level** and with the UI version specified by **@**Version.

Copyright © 2014 Microsoft Corporation.

The Document Version 1 Information and Content Result Set MUST contain one row if it is returned.

```
{Size}                          int,
{SetupPathVersion}              tinyint,
{SetupPath}                     nvarchar(255),
{SetupPathUser}                 nvarchar(255),
{Content}                       image,
{Dirty}                         bit,
DocFlags                        int,
{DoclibRowId}                   int,
{VirusVendorID}                 int,
{VirusStatus}                   int,
{VirusInfo}                     nvarchar(255),
{Version}                       int,
{Id}                          uniqueidentifier,
{Version}                       int;
```

For the field (2) descriptions, see the Document Information and Content (Update) Result Set (section 3.1.5.17.13) above.

3.1.5.17.15 Document Version 2 Information and Content Result Set

The Document Version 2 Information and Content Result Set contains information about the content of the document stream for the nearest available publishing level of the requested version of the document.

The Document Version 2 Information and Content Result Set MUST only be returned if **@GetContent** is set to "1" and **@**Version is not negative, and the document exists with the UI version specified by **@Version**, but with a publishing level visible to the current user that is less than the value returned in the **@Level** output parameter. There MUST be one row in the Document Version 2 Information and Content Result Set if it is returned.

```
{Size}                          int,
SetupPathVersion                tinyint,
SetupPath                       nvarchar(255),
SetupPathUser                   nvarchar(255),
{Content}                       image,
{Dirty}                         bit,
DocFlags                        int,
{DoclibRowId}                   int,
{VirusVendorID}                 int,
{VirusStatus}                   int,
{VirusInfo}                     nvarchar(255),
{Version}                       int,
{DocId}                         uniqueidentifier,
{Version}                       int;
```

For the field (2) descriptions, see the Document Information and Content Result Set (section 3.1.5.16.10) above.

3.1.5.17.16 Attachment State Result Set

The Attachment State Result Set contains information about the attachment state of the requested document. The Attachment State Result Set MUST be returned and MUST contain one row.

Copyright © 2014 Microsoft Corporation.

The Attachment State Result Set is defined in the Common Result Sets Attachment State Result Set (section 2.2.5.2) section.

3.1.5.18 proc_FetchWelcomeNames

The **proc_FetchWelcomeNames** stored procedure lists all the names of the default welcome pages used as redirection targets when folders are requested by an HTTP GET in all site collections in the BEDS.

```
PROCEDURE proc_FetchWelcomeNames ();
```

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

The **proc_FetchWelcomeNames** stored procedure MUST return one result set:

3.1.5.18.1 Welcome Pages Result Set

The Welcome Pages Result Set returns the list of configured default welcome page names, with one row for each configured welcome page name.

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

LeafName: A string containing the configured welcome page name in leaf name format, for example, "default.aspx", "default.htm".

3.1.5.19 proc_GenerateNextId

The **proc_GenerateNextId** stored procedure returns an identifier to be used for a new list item in a specified list (1), and to increment the value of the identifier returned by the next call to **proc_GenerateNextId** for the same List.

```
PROCEDURE proc_GenerateNextId (
      @WebId                      uniqueidentifier,
      @ListId                     uniqueidentifier,
      @NumIds                     int = 1
);
```

@WebId: The Site Identifier (section 2.2.1.11) of the site (2) containing the List.

@ListId: The List Identifier (section 2.2.1.5) of the List.

@NumIds: proc_GenerateNextId MUST increment the value of the list item identifier returned by the next call by this value. This value MUST be a positive number (greater than zero). If the parameter is not provided, 1 is used.

Return Values: The **proc_GenerateNextId** stored procedure returns a positive integer return code, which is the available list item Identifier (section 2.2.1.6) for the specified List.

The **proc_GenerateNextId** stored procedure MUST NOT return a result set.

Copyright © 2014 Microsoft Corporation.

3.1.5.20 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
);
```

@SiteID: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the List.

@WebID: The Site Identifier (section 2.2.1.11) of the site (2) containing the List.

@ListID: The List Identifier (section 2.2.1.5) 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**.

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

The **proc_GetAllAttachmentsInfo** stored procedure MUST return one of two result sets:

3.1.5.20.1 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. 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.

```
      A                             int,
      LeafName                      nvarchar(128),
      Id                            uniqueidentifier,
      Version                       int,
      Acl                           image,
      AnonymousPermMask             bigint;
```

A: 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 (section 2.2.1.2) of the attachment document.

Version: Contains the version of the attachment document.

Acl: Contains the binary serialization of the WSS ACL Format (section 2.2.4.6) access control list for the attachment document. This is either explicitly defined, or inherited from the parent object.

AnonymousPermMask: Contains a WSS Rights Mask (section 2.2.2.13) indicating the rights granted to an anonymous user, or to a user who has no specific rights to the document.

Copyright © 2014 Microsoft Corporation.

3.1.5.20.2 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 (section 2.2.1.2) for the attachment document.

LeafName: Contains the leaf name of the attachment document.

3.1.5.21 proc_GetAuditMask

The **proc_GetAuditMask** stored procedure is invoked to identify Audit Flags (section 2.2.2.1) information for a specified object (a page, file, document, or site collection).

```
PROCEDURE proc_GetAuditMask(
      @ItemType                     tinyint,
      @SiteId                       uniqueidentifier,
      @DirName                      nvarchar(256),
      @LeafName                     nvarchar(128)
);
```

@ItemType: The Audit Item Type (section 2.2.3.2) of the object.

@SiteId: The Site Collection Identifier (section 2.2.1.9) for the site collection containing the object. This is used to get the global audit mask.

@DirName: The directory name of the object.

@LeafName: The leaf name of the object.

Return Values: The **proc_GetAuditMask** stored procedure returns an integer return code which MUST be 0, and the **proc_GetAuditMask** stored procedure MUST return one result set:

3.1.5.21.1 Audit Mask Result Set

The Audit Mask Result Set contains the context-sensitive identifier for the specified object, and the Audit Flags (section 2.2.2.1) set and inherited on that object. The Audit Mask Result Set MUST be returned and MUST contain a single row. The Audit Mask Result Set is defined in the Common Result Sets Site Audit Mask Result Set (section 2.2.5.20).

3.1.5.22 proc_GetAuditMaskOutput

The **proc_GetAuditMaskOutput** stored procedure is invoked to get Audit Flags (section 2.2.2.1) information about an object (a page, file, document, or site collection).

```
PROCEDURE proc_GetAuditMaskOutput(
      @ItemType                     tinyint,
      @SiteId                       uniqueidentifier,
      @DirName                      nvarchar(256),
```


Copyright © 2014 Microsoft Corporation.

```
      @LeafName                  nvarchar(128),
      @Id                        uniqueidentifier          OUTPUT,
      @AuditFlags                int                       OUTPUT,
      @InheritAuditFlags         int                       OUTPUT,
      @GlobalAuditMask           int                       OUTPUT
);
```

@ItemType: The Audit Item Type (section 2.2.3.2) of the object.

@SiteId: A Site Collection Identifier (section 2.2.1.9) for the site collection containing the object. This is used to get the global audit mask information.

@DirName: The directory name of the object.

@LeafName: The leaf name of the object.

@Id: A Document Identifier (section 2.2.1.2) identifying the object, returned as an output parameter.

@Au**d**itFlags**:** An Audit Flags value of the operations to be audited which are set directly on the specified object, returned as an output parameter.

@InheritAuditFlags: An Audit Flags value of the operations to be audited on the specified object which are inherited from its containing list (1), site (2), or site collection, returned as an output parameter.

@GlobalAuditMask: An Audit Flags value of the operations to be audited on the specified object which are inherited from the specified site collection, returned as an output parameter.

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

Value	Description
0	Successful execution.
2	An object of the specified Audit Item Type could not be found at the specified location.

The **proc_GetAuditMaskOutput** stored procedure MUST return no result sets.

3.1.5.23 proc_GetBlobIdsDocs

The **proc_GetBlobIdsDocs** stored procedure is invoked to read a batch of document stream information which is stored in external storage for the current version of each document.

```
PROCEDURE proc_GetBlobIdsDocs(
      @SiteId                    uniqueidentifier,
      @DocId                     uniqueidentifier
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection which contains the documents identified by **@DocId**.

@DocId: The Document Identifier (section 2.2.1.2) of the starting document of the batch to be generated.

Copyright © 2014 Microsoft Corporation.

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

The **proc_GetBlobIdsDocs** stored procedure MUST return one Document Stream External Storage Information Result Set.

3.1.5.23.1 Document Stream External Storage Information Result Set

The Document Stream External Storage Information Result Set MUST contain from 0 up to 1000 rows of document stream information, one row for each document stream found in external storage for the current version of a document with a Document Identifier (section 2.2.1.2) greater than or equal to the value specified by **@DocId**, and MUST be ordered by site collection, Document Identifier and publishing level.

```
        Content                         image,
        ID                              uniqueidentifier,
        {Version}                       int;
```

Content: The document stream content.

ID: The Document Identifier of the document associated with this document stream.

{Version}: This parameter MUST be the constant integer 0.

3.1.5.24 proc_GetBlobIdsVersions

The **proc_GetBlobIdsVersions** stored procedure is invoked to read a batch of document stream information which is stored in external storage for historical versions of each document.

```
PROCEDURE proc_GetBlobIdsVersions (
      @SiteId                       uniqueidentifier,
      @DocId                        uniqueidentifier,
      @Version                      int
);
```

@SiteId: Site Collection Identifier (section 2.2.1.9) of the site collection which contains the documents historical versions.

@DocId: The document identifier (section 2.2.1.2) of the starting document to include in the batch.

@Version: A version number less than any historical version number for the initial document to return. For an initial call to **proc_GetBlobIdsVersions**, this can be 0. Otherwise, to obtain the next batch of information, this can be the version number of the last document returned by a previous call to **proc_GetBlobIdsVersions** before the starting document to include in the batch.

Return Values: The **proc_GetBlobIdsVersions** result set MUST return a return code of 0.

The **proc_GetBlobIdsVersions** result set MUST return one result set as follows:

3.1.5.24.1 Document Version External Storage Information Result Set

The Document Version External Storage Information Result Set MUST contain from 0 up to 1000 rows of document stream information, one row for each document stream found in external storage for the historical versions of documents with a Document Identifier (section 2.2.1.2) greater than or

Copyright © 2014 Microsoft Corporation.

equal to the value specified by **@DocId**, and MUST be ordered by site collection, Document Identifier, and version.

```
    Content                     image,
    ID                          uniqueidentifier,
    Version                     int;
```

Content: The document stream content of the historical version of the document.

ID: The Document Identifier of the document with a historical version. This value MUST be greater than or equal to **@DocId**.

Version: The version number of the historical version of the document. If ID is equal to **@**DocId, then the Version value MUST be greater than **@Version**.

3.1.5.25 proc_GetContainingList

The **proc_GetContainingList** stored procedure is invoked to get metadata and event receiver information about the list (1) containing a specified URL.

```
PROCEDURE proc_GetContainingList(
    @SiteId                     uniqueidentifier,
    @WebId                      uniqueidentifier,
    @Url                        nvarchar(260)
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the List.

@WebId: The Site Identifier (section 2.2.1.11) of the site (2) containing the List.

@Url: The store-relative form URL of a list item or document within the List. The URL is used to derive the location of the containing List. The leaf name part of **@Url** is ignored, since it could point to a nonexistent document or list item.

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

Value	Description
0	Successful execution.
1	List not found in the specified Site.
3	List not found.

The **proc_GetContainingList** stored procedure MUST return one result set if the containing list was not found, and MUST return two result sets if the containing list was found, as defined below:

3.1.5.25.1 List Metadata Result Set

The List Metadata Result Set contains the metadata associated with the List containing the value in **@Url**.

The List Metadata Result Set MUST only be returned if a containing List is found for the value in **@Url**, and MUST contain one row.

Copyright © 2014 Microsoft Corporation.

The List Metadata Result Set is defined in the Common Result Sets List Metadata Result Set (section 2.2.5.12).

3.1.5.25.2 Empty Result Set

The Empty Result Set MUST only be returned if a containing List is not found for the value in **@Url**.

The Empty Result Set is defined in the Common Result Sets Empty Result Set (section 2.2.5.8).

3.1.5.25.3 Event Receivers Result Set

The Event Receivers Result Set contains information about the event receivers defined for the List that contains the value in **@Url**.

The Event Receivers Result Set MUST only be returned if a containing List is found for the value in **@Url**. This result set MUST contain one row for each event receiver registered with an Event Host Type (section 2.2.3.5) of 2 (List) for the List.

The Event Receivers Result Set is defined in the Common Result Sets Event Receivers Result Set (section 2.2.5.9).

3.1.5.26 proc_GetDocsMetaInfo

The **proc_GetDocsMetaInfo** stored procedure is invoked to request Document Metadata information for up to ten Documents within a specified site (2).

```
PROCEDURE proc_GetDocsMetaInfo(
      @DocSiteId                    uniqueidentifier,
      @WebFullUrl                   nvarchar(260),
      @GetDocsFlags                 int,
      @UserId                       int,
      @DirName1                     nvarchar(256)      = NULL,
      @LeafName1                    nvarchar(128)      = NULL,
      @AttachmentsFlag1             tinyint            = NULL,
      @Level1                       tinyint            = NULL,
      @DirName2                     nvarchar(256)      = NULL,
      @LeafName2                    nvarchar(128)      = NULL,
      @AttachmentsFlag2             tinyint            = NULL,
      @Level2                       tinyint            = NULL,
      @DirName3                     nvarchar(256)      = NULL,
      @LeafName3                    nvarchar(128)      = NULL,
      @AttachmentsFlag3             tinyint            = NULL,
      @Level3                       tinyint            = NULL,
      @DirName4                     nvarchar(256)      = NULL,
      @LeafName4                    nvarchar(128)      = NULL,
      @AttachmentsFlag4             tinyint            = NULL,
      @Level4                       tinyint            = NULL,
      @DirName5                     nvarchar(256)      = NULL,
      @LeafName5                    nvarchar(128)      = NULL,
      @AttachmentsFlag5             tinyint            = NULL,
      @Level5                       tinyint            = NULL,
      @DirName6                     nvarchar(256)      = NULL,
      @LeafName6                    nvarchar(128)      = NULL,
      @AttachmentsFlag6             tinyint            = NULL,
      @Level6                       tinyint            = NULL,
      @DirName7                     nvarchar(256)      = NULL,
      @LeafName7                    nvarchar(128)      = NULL,
```


Copyright © 2014 Microsoft Corporation.

```
        @AttachmentsFlag7               tinyint             = NULL,
        @Level7                         tinyint             = NULL,
        @DirName8                       nvarchar(256)       = NULL,
        @LeafName8                      nvarchar(128)       = NULL,
        @AttachmentsFlag8               tinyint             = NULL,
        @Level8                         tinyint             = NULL,
        @DirName9                       nvarchar(256)       = NULL,
        @LeafName9                      nvarchar(128)       = NULL,
        @AttachmentsFlag9               tinyint             = NULL,
        @Level9                         tinyint             = NULL,
        @DirName10                      nvarchar(256)       = NULL,
        @LeafName10                     nvarchar(128)       = NULL,
        @AttachmentsFlag10              tinyint             = NULL,
        @Level10                        tinyint             = NULL
);
```

@DocSiteId: The Site Collection Identifier (section 2.2.1.9) for the site collection containing the specified Documents.

@WebFullUrl: The store-relative form URL of the Site containing the specified Documents.

@GetDocsFlags: A bit mask with a flag that specifies whether to return Link information. If this parameter has the bit 0x00000020 set, then Link information MUST be returned in the Link Info Result Set (section 3.1.5.26.5).

@UserId: The User Identifier (section 2.2.1.13) for the current user requesting the information. This MUST be used by **proc_GetDocsMetaInfo** for permission checking.

The next four parameters are duplicated 10 times, with each set of parameters referring to a document to be fetched. Each instance of these individual parameter names is differentiated by a suffix with a value of 1 thru 10, which replaces the placeholder "#" symbol shown below.

@DirName#: The directory name of the specified document. A NULL value signifies that no document is being fetched in this slot, and the next three parameters MUST be ignored.

@LeafName#: The leaf name of the specified document. If **@DirName#** is not NULL, this MUST NOT be NULL.

@AttachmentsFlag#: An Attachments Flag (section 2.2.3.1) value which specifies the type of security checks to be performed by **proc_GetDocsMetaInfo** based on whether it appears to be an attachment.

@Level#: A Publishing Level Type (section 2.2.2.6) value specifying the maximum publishing level of the current version of the document to be returned to the WFE if multiple Current Versions of the document are available.

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

The **proc_GetDocsMetaInfo** stored procedure MUST return multiple result sets**:**

3.1.5.26.1 Individual URL Security Result Set

The Individual URL Security Result Set contains security information about the specified document. The Individual URL Security Result Set to be used to check if the current user has permission to see this Document's metadata. If the document does not exist, but the specified URL is within a list (1)

Copyright © 2014 Microsoft Corporation.

or document library, security information is returned from the effective scope for the specified document location.

One Individual URL Security Result Set or NULL Individual URL Security Result Set MUST be returned for each document whose corresponding **@DirName#** parameter is not NULL. If all **@DirName#** parameters are set to NULL, Individual URL Security Result Sets MUST NOT be returned.

Each Individual URL Security Result Set MUST only be returned if the specified document location is contained within a List or document library. Otherwise, the NULL Individual URL Security Result Set (section 2.2.5.14) MUST be returned instead. If returned, the Individual URL Security Result Set MUST contain a single row.

The result set is defined in the Common Result Sets Individual URL Security Result Set (section 2.2.5.10).

3.1.5.26.2 NULL Individual Url Security Result Set

The NULL Individual URL Security Result Set indicates that the specified document location is not contained within a List or document library.

One Individual URL Security Result Set or NULL Individual URL Security Result Set MUST be returned for each document whose corresponding **@DirName#** parameter is not NULL. If all **@DirName#** parameters are set to NULL, NULL Individual URL Security Result Sets MUST NOT be returned.

Each NULL Individual URL Security Result Set MUST only be returned if the specified document location is not contained within a List or document library. Otherwise, the Individual URL Security Result Set (section 2.2.5.10) MUST be returned instead. If returned, the NULL Individual URL Security Result Set MUST contain a single row.

The NULL Individual URL Security Result Set is defined in the Common Result Sets NULL Individual URL Security Result Set (section 2.2.5.14).

3.1.5.26.3 Server Time Result Set

The Server Time Result Set returns the current time from the BEDS in UTC. This can be used for a variety of comparisons, and helps handle the case of clock skew between machines in a Farm. The Server Time Result Set MUST be returned. The Server Time Result Set MUST only contain a single row. The Server Time Result Set is defined in the Common Result Sets Server Time Result Set (section 2.2.5.18).

3.1.5.26.4 Subsite List Result Set

The Subsite List Result Set contains an unordered list of store-relative form URLs for all subsites whose parent Site is specified in the **@WebFullUrl** parameter.

The Subsite List Result Set MUST be returned. It MUST contain one row for each subsite with the specified parent Site, and MUST contain no rows if there are no such subsites.

The Subsite List Result Set is defined in the URL Result Set (section 2.2.5.26).

3.1.5.26.5 Link Info Result Set

The Link Info Result Set returns information about each forward link from, and each backward link to, the current version of the specified Documents within the site collection. The Link Info Result Set

Copyright © 2014 Microsoft Corporation.

MUST only be returned if Link information was requested by the **@GetDocsFlags** parameter. The Link Info Result Set MUST contain one row per Link for each specified document.

The Link Info Result Set MUST be ordered by the DocId column, and is defined in the Common Result Sets Single Doc Link Information Result Set (section 2.2.5.19).

3.1.5.26.6 Multiple Document Metadata Result Set

The Multiple Document Metadata Result Set contains the metadata for the specified Documents. The Multiple Document Metadata Result Set MUST contain one row for each document where the corresponding **@DirName#** parameter was not set to NULL, and the rows MUST be ordered by the DocId column.

```
DocId                       uniqueidentifier,
{FullUrl}                   nvarchar(385),
Type                        tinyint,
MetaInfoTimeLastModified    datetime,
MetaInfo                    image,
Size                        int,
TimeCreated                 datetime,
TimeLastModified            datetime,
Version                     int,
DocFlags                    int,
{ListType}                  int,
tp_Name                     nvarchar(38),
{ListTitle}                 nvarchar(255),
{CacheParseId}              uniqueidentifier,
GhostDirName                nvarchar(256),
GhostLeafName               nvarchar(128),
tp_Login                    nvarchar(255),
CheckoutDate                datetime,
{CheckoutExpires}           datetime,
VirusStatus                 int,
VirusInfo                   nvarchar(255),
SetupPathVersion            tinyint,
SetupPath                   nvarchar(255),
SetupPathUser               nvarchar(255),
NextToLastTimeModified      datetime,
UIVersion                   int,
CheckinComment              nvarchar(1023),
WelcomePageUrl              nvarchar(260),
WelcomePageParameters       ntext,
tp_Flags                    bigint,
Acl                         image,
AnonymousPermMask           bigint,
DraftOwnerId                int,
Level                       tinyint,
ParentVersion               int,
TransformerId               uniqueidentifier,
ParentLeafName              nvarchar(128),
ProgId                      nvarchar(255),
DoclibRowId                 int,
tp_DefaultWorkflowId        uniqueidentifier,
ListId                      uniqueidentifier;
```

DocId: If the specified document exists, this MUST be the document identifier (section 2.2.1.2), otherwise this MUST be a generated document identifier (2) value for the specified document.

Copyright © 2014 Microsoft Corporation.

{FullUrl}: The store-relative form URL for the specified document. This value MUST be NULL if the document does not exist. Otherwise, the size of the nvarchar type returned SHOULD vary depending on whether the store-relative form URL has only a directory name, only a leaf name, or both. If it has an empty directory name, it SHOULD be returned as nvarchar(128). If it has an empty leaf name, it SHOULD be returned as nvarchar(256). Otherwise, it MUST be returned as nvarchar(385). Since a WFE client cannot determine which type will be returned in advance, it MUST always allow for a data type returned as nvarchar(385).

Type: The Document Store Type (section 2.2.2.4) of this document. This value MUST be NULL if the requested document does not exist.

MetaInfoTimeLastModified: A datetime with a timestamp in UTC format specifying the last time the Metainfo value of this document was changed. This value MUST be NULL if the document does not exist.

MetaInfo: A metadict for the document. The metadict format is specified in [MS-FPSE] section 2.2.2.2.11. This value MUST be NULL if the document does not exist.

Size: The number of bytes in the document stream. This value MUST be NULL if the document does not exist.

TimeCreated: A datetime with a timestamp in UTC format specifying when this document was created. This value MUST be NULL if the document does not exist.

TimeLastModified: A datetime with a timestamp in UTC format specifying when the document was last modified. This corresponds to the TimeCreated (for historical versions) or TimeLastModified (for Current Versions) of the document. This value MUST be NULL if the document does not exist.

Version: A counter incremented any time a change is made to this document, used for internal conflict detection. This value MUST be NULL if the document does not exist.

DocFlags: A Doc Flags (section 2.2.2.3) value describing the document. This value MUST be NULL if the document does not exist.

{ListType}: A combination of the Base Type and Server Template values of the associated List for this document, where the Server Template value is multiplied by 256 and added to the value of the Base Type. This value MUST be NULL if the document does not exist or if it is not in a List.

tp_Name: The Identifier of the List that contains this document. This value MUST be NULL if the document does not exist or if it is not in a List.

{ListTitle}: If this document is the root folder of a List, this contains the display name of the List. This value MUST be NULL if the document does not exist or if it is not in a List.

{CacheParseId}: This value MUST be NULL.

GhostDirName: The directory name as passed to **proc_GetDocsMetaInfo** in the **@DirName#** parameter.

GhostLeafName: The leaf name as passed to **proc_GetDocsMetaInfo** in the **@LeafName#** parameter.

tp_Login: If the document exists and is currently checked out, **tp_Login** is the login name of the User to whom it is checked out. In all other cases, this is NULL.

Copyright © 2014 Microsoft Corporation.

CheckoutDate: A datetime with a timestamp in UTC format specifying when this document was checked out. If the document is currently checked in or the document does not exist, this MUST be NULL.

{CheckoutExpires}: A datetime with a timestamp in UTC format specifying when the short-term lock for this document will expire. If this date is in the past, the document SHOULD be treated as unlocked. This value can be NULL if no short-term lock has been placed on the document. This MUST be NULL if a User has the document checked out, or if the document is not currently locked with a short-term lock, or if it does not exist.

VirusStatus: A Virus Status (section 2.2.3.18) value specifying the current virus state of this document. This value can be NULL if the document has not been processed by a virus scanner. This value MUST be NULL if the document does not exist.

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 the document does not exist or if the document has not been processed by a virus scanner.

SetupPathVersion: For a ghosted document, this specifies the setup path location that the SetupPath fragment is relative to. This value MUST NOT be NULL. The following are valid values:

Value	Description
2	This is relative to the install location of Windows SharePoint Services 2.0 on the WFE (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\60)
3	This is relative to the install location of Windows SharePoint Services 3.0 on the WFE (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\12).

SetupPath: For a document that is or has been ghosted, this contains the setup path fragment relative to the base setup path specified by the **SetupPathVersion** value, where the content stream of this document can be found. This value MUST be NULL for a document that does not exist or was never ghosted.

SetupPathUser: If this document is now or once was ghosted, this contains the login name of the User that created the ghosted document. This value MUST be NULL for a document that does not exist or was never ghosted.

NextToLastTimeModified: The value of TimeLastModified from the previous time when the document was last saved. This value MUST be NULL for a document that does not exist.

UIVersion: The UI version number of the document. This value MUST be NULL for a document that does not exist.

CheckinComment: An optional user-supplied description provided when a document is checked in or published. This value MUST be NULL for a document that does not exist or is not checked in.

WelcomePageUrl: If this document is a folder, this specifies an optional page to redirect to when the folder is requested with an HTTP GET operation. The URL is relative to the URL of the folder itself, and MUST be contained within that folder. Attempts to break out of the folder, such as "../../somepage.aspx", are not valid. If the document does not exist, or a welcome page is not specified, this value MUST be NULL.

WelcomePageParameters: Contains optional URL "search" parameters to specify as part of the **WelcomePageUrl** value. These parameters are the URL substring starting at either the query string signifier '?' or the bookmark signifier '#'. If the document does not exist, or parameters are not specified, this value MUST be NULL.

Copyright © 2014 Microsoft Corporation.

tp_Flags: The List Flags (section 2.2.2.5) value for the List that contains this document. This value MUST be NULL for Documents that do not exist or are not in a List.

Acl: The binary serialization of the WSS ACL Format (section 2.2.4.6) access control list for this document. The WSS ACL is either explicitly defined for the document, or inherited from the parent object of the document. This value MUST be NULL for Documents that do not exist.

AnonymousPermMask: A WSS Rights Mask indicating the rights granted to an anonymous user, or to a user who has no specific rights to this document. The value MUST be NULL for Documents that do not exist.

DraftOwnerId: The User Identifier (section 2.2.1.13) of the User that published this document as a draft. This value MUST be NULL if the document is not a draft version or does not exist.

Level: A Publishing Level Type (section 2.2.2.6) value specifying the publishing status of this document. This value MUST be NULL if the document does not exist.

ParentVersion: If the document is a transformed version of another document, this is the UI version value from the parent document. This value MUST be NULL if the document does not exist or is not the product of a document transformation.

TransformerId: If this document is a transformed version of another document, this is the Globally Unique Identifier (GUID) of the agent that performed the transformation. This value MUST be NULL if the document does not exist or is not the product of a document transformation.

ParentLeafName: If the document is a transformed version of another document, this is the leaf name of the original document. The original document MUST be in the same folder as the transformed document. If either is moved, the relationship is broken. This value MUST be NULL if the document is not the product of a document transformation or does not exist.

ProgId: Specifies a preferred application to open the document. The **ProgId** value is used to distinguish between different applications that save files with a given file extension (for example, different editors for .HTML or .XML files). This value MUST be NULL if the document does not exist or if a **ProgId** was not specified when the document was saved.

DoclibRowId: The document library row identifier for this item. If the document does not exist or is not contained in a List, this value MUST be NULL.

tp_DefaultWorkflowId: The Workflow Identifier (section 2.2.1.16) corresponding to the workflow (2) to be invoked if the document is in a moderated List and the document is submitted for approval as part of a check in. If the document does not exist or is not contained in a List with a configured approval workflow (2), this value MUST be NULL.

ListId: The List Identifier (section 2.2.1.5) of the List that contains the document. If the document does not exist or is not contained in a List, this value MUST be NULL.

3.1.5.27 proc_GetLinkInfoSingleDoc

The **proc_GetLinkInfoSingleDoc** stored procedure provides link information and status for all forward links within a specified document, and for all backward links within the specified site collection to the document.

```
PROCEDURE proc_GetLinkInfoSingleDoc(
      @DocSiteId                    uniqueidentifier,
      @DocDirName                   nvarchar(256),
      @DocLeafName                  nvarchar(128),
      @Level                        tinyint
```


Copyright © 2014 Microsoft Corporation.

```
);
```

@DocSiteId: The Site Collection Identifier (section 2.2.1.9) for a site collection containing the document.

@DocDirName: The directory name of the document.

@DocLeafName: The leaf name of the document.

@Level: A Publishing Level Type (section 2.2.2.6) value indicating the publishing level of the document.

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

The **proc_GetLinkInfoSingleDoc** stored procedure MUST return a single result set, as specified in the following sections.

3.1.5.27.1 Link Info Single Doc Result Set

The Link Info Single Doc Result Set contains information about Links to or within the requested document. Entries are present both for forward links and for backward links . The Link Info Single Doc 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 Link Info Single Doc Result Set is defined in the Common Result Sets Link Information Result Set (section 2.2.5.11).

3.1.5.28 proc_GetListCheckedOutFiles

The **proc_GetListCheckedOutFiles** stored procedure is invoked to retrieve a list of all documents with a Document Store Type (section 2.2.2.4) of 0 (File) within a specified list (1), folder, or its subfolders, which are checked out and do not have checked in draft or published versions.

```
PROCEDURE proc_GetListCheckedOutFiles(
      @SiteId                       uniqueidentifier,
      @ListUrl                      nvarchar(260)
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the List.

@ListUrl: The directory name which contains the document(s), preceded with a leading slash ("/").

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

The **proc_GetListCheckedOutFiles** stored procedure MUST return one result set.

3.1.5.28.1 Checked Out Files Result Set

The Checked Out Files Result Set returns information in the form defined below about the documents which are checked out. The Checked Out Files Result Set MUST return one row for each document with a Document Store Type (section 2.2.2.4) of 0 (File) in the specified directory which is checked out and does not have a checked in draft or published version.

Copyright © 2014 Microsoft Corporation.

```
DirName                         nvarchar(256),
LeafName                        nvarchar(128),
DocLibRowId                     int,
CheckoutUserId                  int,
{tp_Title}                      nvarchar(255),
{tp_Email}                      nvarchar(255),
TimeLastModified                datetime,
Size                            int;
```

DirName: The directory name of the Directory containing the document(s).

LeafName: The leaf name of the document.

DocLibRowId: The row identifier for the document within the containing document library.

CheckoutUserId: The User Identifier (section 2.2.1.13) for the principal (1) who has this document checked out.

{tp_Title}: The display name of the principal (1) specified in **CheckoutUserId**. This parameter MUST NOT be NULL. If display name is unavailable, **{tp_Title}** MUST contain an empty string.

{tp_Email}: The email address of the User specified in **CheckoutUserId**. This parameter MUST NOT be NULL. If the email address is unavailable, **{tp_Email}** MUST contain an empty string.

TimeLastModified: A date/time value in UTC format specifying when the document was last saved.

Size: The size of the document in bytes.

3.1.5.29 proc_GetListFields

proc_GetListFields is invoked to get the mapping of fields in a list (1). The mapping is represented via an XML string specifying each of the fields in the list (1).

```
PROCEDURE proc_GetListFields(
      @WebId                        uniqueidentifier,
      @ListId                       uniqueidentifier
);
```

@WebId: The Site Identifier (section 2.2.1.11) of the site (2) containing the list (1).

@ListId: The List Identifier (section 2.2.1.5) of the list (1).

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

The **proc_GetListFields** stored procedure MUST return a single result set as defined below:

3.1.5.29.1 Fields Information Result Set

The Fields Information Result Set MUST return a single row holding a single column. The Fields Information Result Set will be returned once, and MUST contain 0 or 1 rows as follows: if no List was found matching the provided **@WebId** and **@ListId** parameters, the result set MUST contain 0 rows. Otherwise, the result set MUST contain one row.

Copyright © 2014 Microsoft Corporation.

```
        tp_Fields                       ntext;
```

tp_Fields: Contains a WSS implementation-specific version string followed by an XML fragment representation of the field (2) definitions of the list (1), as specified in the **FieldDefinitionDatabaseWithVersion** type defined in section 2.2.8.3.5.

3.1.5.30 proc_GetListMetaDataAndEventReceivers

proc_GetListMetaDataAndEventReceivers is invoked to retrieve information about the metadata, scopes, web parts and event receivers for a specified list (1).

```
PROCEDURE proc_GetListMetaDataAndEventReceivers(
      @SiteId                       uniqueidentifier,
      @WebId                        uniqueidentifier,
      @ListId                       uniqueidentifier,
      @PrefetchListScope            bit               =0,
      @PrefetchWebParts             bit               =0,
      @UserId                       int               =-1
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) for a site collection that contains **@WebId**.

@WebId: The Site Identifier (section 2.2.1.11) for the site (2) containing the list (1).

@ListId: The List Identifier (section 2.2.1.5) for the list (1).

@PrefetchListScope: Specifies whether to include the Unique Permissions Result Set or NULL Unique Permissions Result Set in the returned result sets. If 1, then the Unique Permissions Result Set or NULL Unique Permissions Result Set MUST be returned. Otherwise, the Unique Permissions Result Set and NULL Unique Permissions Result Set MUST NOT be returned.

@PrefetchWebParts: Specifies whether to include the List Web Parts Result Set in the returned result sets. If 1, then the List Web Parts Result Set MUST be returned. Otherwise, the List Web Parts Result Set MUST NOT be returned.

@UserId: User Identifier (section 2.2.1.13). This value can refer to an existing user identifier for the specified site collection, and defaults to -1, specifying that the contents of the List Web Parts Result Set MUST NOT be limited to results for a particular User. When an existing User Identifier is specified, the contents of the List Web Parts Result Set MUST be limited to web parts visible to the specified User.

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

The **proc_GetListMetaDataAndEventReceivers** stored procedure MUST return between two and four result sets, depending upon input parameters. result sets that are returned will be sent in the order defined below**:**

3.1.5.30.1 List Metadata Result Set

The List Metadata Result Set contains the metadata associated with the list (1). The List Metadata Result Set MUST be returned and MUST contain one row for a valid List Identifier (section 2.2.1.5) in the **@ListID** parameter. If the List Identifier is invalid, the List Metadata Result Set MUST contain zero rows.

Copyright © 2014 Microsoft Corporation.

The List Metadata Result Set is defined in the Common Result Sets List Metadata Result Set (section 2.2.5.12).

3.1.5.30.2 Unique Permissions Result Set

The Unique Permissions Result Set returns any scopes associated with list items or folders within the specified list (1). The Unique Permissions Result Set MUST be returned when the **@PrefetchListScope** parameter is set to 1, if at least one such scope exists. Otherwise, the NULL Unique Permissions Result Set MUST be returned to indicate no such scopes exist. The Unique Permissions Result Set MUST contain one row for each scope defined on any list item or folder within the list (1), but no more than 1001 rows.

The Unique Permissions Result Set is defined in the Common Result Sets Unique Permissions Result Set (section 2.2.5.25).

3.1.5.30.3 NULL Unique Permissions Result Set

The NULL Unique Permissions Result Set indicates that no unique scopes exist within the specified list (1). The NULL Unique Permissions Result Set MUST be returned if the **@PrefetchListScope** parameter is set to 1 and no scopes are defined for any list items or folders within the specified list.

The NULL Unique Permissions Result Set is defined in the Common Result Sets NULL Unique Permissions Result Set (section 2.2.5.15).

3.1.5.30.4 List Event Receivers Result Set

The List Event Receivers Result Set contains information about the event receivers defined for this list (1). The List Event Receivers Result Set MUST be returned, and MUST contain one row for each event receiver that is registered for this list (1), or zero rows if no event receivers exist.

The List Event Receivers Result Set is defined in the Common Result Sets Event Receivers Result Set (section 2.2.5.9).

3.1.5.30.5 List Web Parts Result Set

The List Web Parts Result Set contains information about the List web parts defined for this site (2). The List Web Parts Result Set MUST be returned if the **@PrefetchWebParts** parameter is set to 1. The result set MUST contain one row for each web part that is registered for this list (1).

The List Web Parts Result Set is defined in the Common Result Sets List Web Parts Result Set (section 2.2.5.13).

3.1.5.31 proc_getObject

The **proc_getObject** stored procedure is invoked to retrieve a single Configuration Object (section 2.2.6.1) from the database.:

```
PROCEDURE proc_getObject(
      @Id                           uniqueidentifier,
);
```

@Id: Contains the GUID for the Id of the Configuration Object to be retrieved.

Return Values: Upon successful execution, the **proc_getObject** stored procedure returns an integer return code of 0.

Copyright © 2014 Microsoft Corporation.

The **proc_getObject** stored procedure MUST return a single result set as follows:

3.1.5.31.1 Object Result Set

The Object Result Set returns the Configuration Object (section 2.2.6.1) identified by **@Id**. If **@Id** is NULL or does not match the value of any Configuration Objects in the Configuration Database, the Object Result Set MUST be returned and MUST return zero rows.

The Object Result Set MUST be returned and MUST return one row if one Configuration Object matches the value of **@Id**.

```
Id                              uniqueidentifier,
ParentId                        uniqueidentifier,
ClassId                         uniqueidentifier,
Name                            nvarchar(128),
Status                          int,
Version                         rowversion,
Properties                      ntext;
```

Id: Contains the GUID for the Id of the requested Configuration Object.

ParentId: Contains the GUID for the **ParentId** of the Configuration Object.

ClassId: Contains the GUID for the **ClassId** of the Configuration Object.

Name: Contains the Name of the Configuration Object.

Status: Contains the Status of the Configuration Object.

Version: Contains the version of the Configuration Object.

Properties: Contains the Properties of the Configuration Object .

3.1.5.32 proc_getObjectsByBaseClass

The **proc_getObjectsByBaseClass** stored procedure is invoked to return a list of GUIDs for child Configuration Objects (section 2.2.6.1) of the specified parent Configuration Object that are inherited from the specified base Class (section 2.2.6.1.1). The **proc_getObjectsByBaseClass** is located in the Configuration Database.

```
PROCEDURE proc_getObjectsByBaseClass (
      @BaseClassId                  uniqueidentifier,
      @ParentId                     uniqueidentifier
);
```

@BaseClassId: Contains the GUID for the base Class of the requested Configuration Objects.

@ParentId: Contains the GUID for the Parent (section 2.2.6.1.4) Configuration Object of the requested Configuration Objects.

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

The **proc_getObjectsByBaseClass** stored procedure MUST return a single result set as follows:

Copyright © 2014 Microsoft Corporation.

3.1.5.32.1 Object ID Result Set

The Object ID Result Set contains the identifiers of the matching Configuration Objects (section 2.2.6.1), if any. The Object ID Result Set MUST contain zero rows if no matching Configuration Objects are found, and MUST contain one row for each matching Configuration Object found. The Object ID Result Set is defined in the Common Result Sets Object ID Result Set (section 2.2.5.16).

3.1.5.33 proc_getObjectsByClass

The **proc_getObjectsByClass** stored procedure is invoked to retrieve the GUID of a Configuration Object (section 2.2.6.1) from the database based upon the Class (section 2.2.6.1.1) type, the Parent (section 2.2.6.1.4) Configuration Object, and the Name (section 2.2.6.1.3) of the Configuration Object. The **proc_getObjectsByClass** is located in the Configuration Database.

```
PROCEDURE proc_getObjectsByClass(
      @ClassId                        uniqueidentifier,
      @ParentId                       uniqueidentifier,
      @Name                           nvarchar(128)
);
```

@ClassId: The GUID of the Class type. MUST NOT be NULL.

@ParentId: The GUID of the Parent Configuration Object. MAY be NULL.

@Name: The Name of the Configuration Object. MAY be NULL.

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

Value	Description
0	Successful execution.
50105	Error: Class does not exist.

The **proc_getObjectsByClass** stored procedure MUST return a single result set as follows:

3.1.5.33.1 Object ID Result Set

The Object ID Result Set contains the GUID of the matching Configuration Object (section 2.2.6.1), if any. The Object ID Result Set MUST contain zero rows if no matching Configuration Object is found, and MUST contain one row if a matching Configuration Object is found. Matching MUST be done on the value specified by **@ClassId**, and on the values specified in **@ParentId** and @Name if not NULL. The Object ID Result Set is defined in the Common Result Sets Object ID Result Set (section 2.2.5.16).

3.1.5.34 proc_GetSiteFlags

The **proc_GetSiteFlags** stored procedure is invoked to return the Site Collection Flags (section 2.2.2.9) set for a specified site collection.

```
PROCEDURE proc_GetSiteFlags(
      @WebSiteId                      uniqueidentifier
);
```


Copyright © 2014 Microsoft Corporation.

@WebSiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection to retrieve Site Collection Flags for.

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

The **proc_GetSiteFlags** stored procedure MUST return the following result set:

3.1.5.34.1 Site Collection Flags Result Set

The Site Collection Flags Result Set returns the Site Collection Flags (section 2.2.2.9) for a specified site collection. The Site Flags Result Set will always be returned, and MUST contain one row.

```
{SiteCollectionFlags}          int;
```

{SiteCollectionFlags}: Contains the Site Collection Flags for the specified site collection. This will be NULL if the input **@WebSiteId** is invalid.

3.1.5.35 proc_getSiteMap

The **proc_getSiteMap** stored procedure is invoked to determine the Site Map information for a site collection. **proc_getSiteMap** is located in the Configuration Database.

```
PROCEDURE proc_getSiteMap(
      @ApplicationId                uniqueidentifier,
      @Path                         nvarchar(128)
);
```

@ApplicationId**: @**ApplicationId**:** The GUID of the web application (1) containing the site collection that exists in the specified Path.

@Path: The server-relative path to the root of the site collection. Set **@Path** to '/' to specify a site collection which is the root of the web application (1).

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

The **proc_getSiteMap** stored procedure MUST return a single result set defined as follows:

3.1.5.35.1 Site Map Result Set

The Site Map Result Set returns the Site Map information for the site collection specified by the **@ApplicationId** and **@Path** parameter values. The Site Map Result Set MUST return zero rows if the parameters do not match any Site Map information; it MUST return one row if a match is found.

```
      Id                            uniqueidentifier,
      DatabaseId                    uniqueidentifier,
      RedirectUrl                   nvarchar(512),
      Pairing                       tinyint;
```

Id: Contains the Site Collection Identifier (section 2.2.1.9) for the requested site collection.

Copyright © 2014 Microsoft Corporation.

DatabaseId: Contains the GUID of the database containing the content of the requested site collection.

RedirectUrl: Contains the URL used to access the requested site collection. This value MUST be NULL if the Site is upgraded to the current version.

Pairing: Contains an implementation-specific integer value denoting that a given object is paired with a matching object of a previous version. This parameter is used during side-by-side version-to-version upgrades.

Value	Description
0	This object is not paired.
1	This object is paired with an object from a previous major version of the product.

3.1.5.36 proc_getSiteMapById

The **proc_getSiteMapById** stored procedure is invoked to determine complete Site Map information for a site collection. This Site Map information includes identifiers for the web application (1) and the database the site collection belongs to, the URL mapped to the site collection, a boolean representing whether or not the site collection is configured to use a host header, and the pairing bit and redirect URL used when the site collection is being upgraded.

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

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection to get information for.

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

The **proc_getSiteMapById** stored procedure MUST return a single result set as follows:

3.1.5.36.1 Site Map By Id Result Set

The Site Map By Id Result Set returns information about the web application (1), database, and URL mapped to the specified site collection. The Site Map By Id Result Set MUST contain one row if the specified Site Collection Identifier (section 2.2.1.9) exists in the Configuration Database on the BEDS. Otherwise, it MUST contain zero rows.

```
      ApplicationId                 uniqueidentifier,
      DatabaseId                    uniqueidentifier,
      Id                            uniqueidentifier,
      Path                          nvarchar(128),
      RedirectUrl                   nvarchar(512),
      Pairing                       tinyint,
      HostHeaderIsSiteName          bit;
```

ApplicationId: The GUID of the web application (1) hosting the content of the requested site collection.

DatabaseId: The GUID of the database containing the content of the requested site collection.

Copyright © 2014 Microsoft Corporation.

Id: The Site Collection Identifier for the requested site collection.

Path: The URL used to identify the requested site collection. For site collections that do not use a specifically-configured host domain, the **HostHeaderIsSiteName** field (2) MUST be 0 and the Path value MUST be the server-relative URL of the root site (2) of the site collection. For site collections using host header mode (that is, site collections identified by the host header value in the requests used to access them, rather than by their server-relative URL path), the **HostHeaderIsSiteName** field (2) MUST be 1 and the Path value for the site collection MUST be the host name, and MUST include the port number for hosts at non-default ports (for example, host.contoso.com:8080), and can include the port number for hosts at default ports.

RedirectUrl: The URL used to access the specified site collection during an implementation-specific upgrade operation. This value MUST be NULL if the site collection upgrade to the current WSS version is complete.

Pairing: A Windows SharePoint Services implementation-specific integer value used during side-by-side version-to-version Windows SharePoint Services upgrade to denote that a given WSS object is paired with a matching object of a previous Windows SharePoint Services version. The following are valid values:

Value	Description
0	This object is not paired.
1	This object is paired with an object from a previous major version of the Windows SharePoint Services product.

HostHeaderIsSiteName: This value MUST be set to 1 if the requested site collection uses host header mode; otherwise it MUST be 0. The Path field (2) MUST contain the appropriate URL for the specified **HostHeaderIsSiteName** value.

3.1.5.37 proc_GetTpWebMetaDataAndListMetaData

proc_GetTpWebMetaDataAndListMetaData is invoked to retrieve metadata for a particular site (2) or list (1).

```
PROCEDURE proc_GetTpWebMetaDataAndListMetaData(
      @WebSiteId                    uniqueidentifier,
      @WebId                        uniqueidentifier,
      @Url                          nvarchar(260),
      @ListId                       uniqueidentifier,
      @RunUrlToWebUrl               bit,
      @DGCacheVersion               bigint,
      @SystemId                     varbinary(512)     = NULL,
      @MetadataFlags                int                = 0
);
```

@WebSiteId: The Site Collection Identifier (section 2.2.1.9) for a site collection containing the site (2).

@WebId: The Site Identifier (section 2.2.1.11) of the site (2) for which metadata is requested, or of the site containing the list (1) for which metadata is requested. This parameter is ignored if the **@MetadataFlags** has the METADATA_WEB flag set and if values in **@WebSiteId** and **@URL** are valid.

Copyright © 2014 Microsoft Corporation.

@Url: The store-relative form URL of the site (2) or list (1) to retrieve metadata for.

@ListId: The List Identifier (section 2.2.1.5) of the List for which metadata is requested. This parameter is optional and MUST be NULL if no List metadata is required.

@RunUrlToWebUrl: A bit flag specifying whether the **@URL** parameter is for a site (2) or for a list (1) within a site. If this value is set to "1", the value in **@URL** is for a list (1), and MUST be converted to a store-relative form URL for the containing site (2).

@DGCacheVersion: The version number of the Domain Group Map Cache in the WFE. This can be compared with the Domain Group Map Cache version number on the BEDS returned in the Domain Group Cache Versions Result Set (section 3.1.5.37.2) to determine whether updates are needed. A **@DGCacheVersion** value of -2 specifies that information about the Domain Group Map Cache is not requested, and the Domain Group Cache BEDS Update Result Set (section 3.1.5.37.3) and the Domain Group Cache WFE Update Result Set (section 3.1.5.37.4) MUST NOT be returned.

@SystemId: The **SystemID** (section 2.2.1.12) of the current user requesting the metadata.

@MetadataFlags: A bit field (2) used to determine the types of metadata returned by this procedure. The only valid values of the Metadata Flags bits are specified as follows:

Symbolic name, Value	Description
METADATA_NONE, 0x00000000	Do not provide any metadata.
METADATA_FETCH_VIEWS, 0x00000001	Retrieve List Views. This flag is set in combination with METADATA_LISTMETADATA_NOFETCH and the @ListID variable and will result in the List Web Parts Result Set (1) (section 3.1.5.37.20).
METADATA_PREFETCH_SCOPES, 0x00000002	Retrieve security scopes. If this flag is set and @ListId is not NULL and Metadata_ListMetadata_NoFetch is not set, the Unique Permissions Result Set (2) (section 3.1.5.37.17) will be returned. The Unique Permissions Result Set (3) (section 3.1.5.37.22) is also returned when set in combination with METADATA_USERINFOLIST and METADATA_USERINFOLIST_NOFETCH is not set, and ListID is NULL or is not equal to the UserInfo ListID.
METADATA_SAVE_SCOPE_DATA, 0x00000004	Save scope data. This flag is currently unused.
METADATA_FIGURE_TYPE, 0x00000008	Setting this flag in combination with NULL ListID will result in retrieving the ListID values directly from the Docs (section 2.2.7.4) table.
METADATA_WEB, 0x00000010	Retrieve information pertaining to the provided Site information. This flag is used in combination with other flags to further define the set or sets of information to be retrieved.
METADATA_URL, 0x00000020	Setting this flag will populate the ListId column for all results, including those which are not list items. When used in combination with a valid **@WebSiteId**, if **@ListId** is NULL or doesn't match the List contained within the Site identified by **@WebSiteId**, the Document Metadata Result Set is returned.

Copyright © 2014 Microsoft Corporation.

Symbolic name, Value	Description
METADATA_USERINFOLIST, 0x00000040	Return User information. Setting this flag when METADATA_USERINFOLIST_NOFETCH is not set opens the option for List Metadata Result Set, List Permissions Result Set, Event Receivers Result Set, and the List Web Parts Result Set.
METADATA_USERINFOLIST_FULL, 0x00000080	Return full User information. Setting this flag in combination with METADATA_USERINFOLIST when METADATA_USERINFOLIST_NOFETCH, is not set, along with a valid Site Identifier in **@WebSiteId**, where **@ListId** is NULL or doesn't match a list identifier contained within the Site specified by **@WebSiteId**, returns the Event Receivers Result Set and the List Web Parts Result Set.
METADATA_WEB_PROP, 0x00000100	Return web data. Setting this flag in combination with METADATA_WEB will retrieve the Web Category Result Set and Web Metainfo Result Set.
METADATA_WEB_FEATURES, 0x00000200	Return Web features. Setting this flag in combination with METADATA_WEB will retrieve the Web Feature List Result Set.
METADATA_WEB_NAVSTRUCT, 0x00000400	Return Navigation Structure. Setting this flag in combination with METADATA_WEB will retrieve the Unique Permissions Result Set.
METADATA_LISTMETADATA_NOFETCH, 0x00000800	Do not return List metadata. Setting this flag will suppress the retrieval of the List Metadata Result Set, Unique Permissions Result Set, Event Receivers Result Set and the List Web Parts Result Set using the List identification parameters.
METADATA_USERINFOLIST_NOFETCH, 0x00001000	Do not return User information. Setting this flag will suppress the retrieval of the List Metadata Result Set, Unique Permissions Result Set, Event Receivers Result Set and the List Web Parts Result Set using the User identification parameters.
METADATA_URL_REALLY, 0x00002000	Return single document metadata. Setting this flag will return the Document Metadata Result Set.
METADATA_WEB_WELCOMEPAGE, 0x00004000	Redirect welcome page. Setting this flag in combination with METADATA_WEB will retrieve the Redirect URL Result Set.

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

Value	Description
0	Successful operation.
1	Cross Site attack detected.
2	The attempt to retrieve a redirected URL failed.
3	The specified URL was not found.
1168	The site collection specified by @WebSiteId was not found.
1271	Access to the site (2) is blocked because the site collection is locked.

Copyright © 2014 Microsoft Corporation.

The **proc_GetTpWebMetaDataAndListMetaData** stored procedure returns up to 27 result sets, depending on the state of the input parameters.

3.1.5.37.1 Web Url Result Set

The Web Url Result Set contains the store-relative form URL of the root of the Site which contains the **@Url** parameter.

The Web Url Result Set MUST be returned when **@MetadataFlags** has the METADATA_WEB flag set, **@RunUrlToWebUrl** is set to 1, and **@WebSiteId** is a valid Site Collection Identifier (section 2.2.1.9), and MUST contain one row of data. If **@MetadataFlags** has the METADATA_WEB flag set, **@RunUrlToWebUrl** is set to 1, and **@WebSiteId** is not found, **proc_GetTpWebMetaDataAndListMetaData** MUST NOT return any result sets.

```
{WebUrl}                    nvarchar(260);
```

{WebUrl}: The store-relative form URL of the Site containing the **@Url** parameter.

3.1.5.37.2 Domain Group Cache Versions Result Set

The Domain Group Cache Versions Result Set contains information about the version numbers associated with the Domain Group Map Caches on the WFE and on the BEDS for the specified site collection.

The Domain Group Cache Versions Result Set MUST be returned when **@Url** specifies a valid location within or of a site (2) and **@MetadataFlags** has the METADATA_WEB flag set, and MUST contain one row of version number data. If **@MetadataFlags** has the METADATA_WEB flag set, and **@Url** does not specify either a valid site, or if **@RunUrlToWebUrl** is set to 1, a valid location within a site, then **proc_GetTpWebMetaDataAndListMetaData** MUST NOT return this or any further result sets.

If the specified **@DGCacheVersion** value is -2, then all columns returned MUST have the value -2, indicating that the value MUST NOT be used for comparison. The Domain Group Cache Versions Result Set is defined in the Common Result Sets Domain Group Cache Versions Result Set (section 2.2.5.4).

3.1.5.37.3 Domain Group Cache BEDS Update Result Set

The Domain Group Cache BEDS Update Result Set contains information to be used in recomputing the domain group map cache, which contains the mapping of domain groups to the site groups they are members of. The presence of The Domain Group Cache BEDS Update Result Set means the database's copy of the domain group map cache is out of date and MUST be recomputed to ensure that proper security checks can be made.

The Domain Group Cache BEDS Update Result Set MUST be returned only if the Domain Group Cache Versions Result Set is returned, **@DGCacheVersion** does not contain "-2", and the value of RealVersion is greater than the CachedVersion in the results in the Domain Group Cache Versions Result Set.

The Domain Group Cache BEDS Update Result Set schema is defined in the Common Result Sets Domain Group Cache BEDS Update Result Set (section 2.2.5.3).

Copyright © 2014 Microsoft Corporation.

3.1.5.37.4 Domain Group Cache WFE Update Result Set

The Domain Group Cache WFE Update Result Set contains the binary data needed to refresh the domain group map cache. The presence of the Domain Group Cache WFE Update Result Set means the domain group map cache on the database is up to date, and the WFE cache can be refreshed if necessary.

The Domain Group Cache WFE Update Result Set MUST be returned only if the Domain Group Cache Versions Result Set is returned, and **@DGCacheVersion** does not contain "-2", and the value of RealVersion is less than or equal to CachedVersion in the results in Domain Group Cache Versions Result Set. The Domain Group Cache WFE Update Result Set MUST contain one row of data.

The Domain Group Cache WFE Update Result Set schema is defined in the Common Result Sets Domain Group Cache WFE Update Result Set (section 2.2.5.5).

3.1.5.37.5 Site Metadata Result Set

The Site Metadata Result Set contains specialized site metadata. The Site Metadata Result Set MUST be returned when the Domain Group Cache Versions Result Set is returned, and it MUST contain one row.

The Site Metadata Result Set is defined in the Common Result Sets Site Metadata Result Set (section 2.2.5.23).

3.1.5.37.6 Event Receivers Result Set (1)

The Event Receivers Result Set contains information about the event receivers defined for this site (2). The Event Receivers Result Set MUST be returned when the Domain Group Cache Versions Result Set is returned, and it MUST contain one row for each event receiver registered for this site. If this Event Receivers Result Set is returned, and the site collection specified by **@WebSiteId** is locked, then **proc_GetTpWebMetaDataAndListMetaData** MUST NOT return any further result sets.

The Event Receivers Result Set is defined in the Common Result Sets Event Receivers Result Set (section 2.2.5.9).

3.1.5.37.7 Site Categories Result Set

The Site Categories Result Set contains the categories of the Site. The Site Categories Result Set MUST be returned when execution is successful to this point, and **@MetadataFlag** has the METADATA_WEB flag set and the METADATA_WEB_PROP flag set.

The Site Categories Result Set is defined in the Common Result Sets Site Categories Result Set (section 2.2.5.22).

3.1.5.37.8 Site MetaInfo Result Set

The Site MetaInfo Result Set contains the metadict of the site (2). The Site MetaInfo Result Set MUST be returned when execution is successful to this point, and **@MetadataFlag** has the METADATA_WEB flag set and the METADATA_WEB_PROP flag set, and MUST contain one row if the **@WebId** parameter is valid. If the **@MetadataFlag** has the METADATA_WEB flag set and the METADATA_WEB_PROP flag set, and the site (2) specified by **@WebId** does not exist, then the result set MUST contain no rows, and **proc_GetTpWebMetaDataAndListMetaData** MUST NOT return any further result sets.

Copyright © 2014 Microsoft Corporation.

The Site MetaInfo Result Set is defined in the Common Result Sets Site MetaInfo Result Set (section 2.2.5.24).

3.1.5.37.9 Site Feature List Result Set

The Site Feature List Result Set contains the list of Feature Identifiers (section 2.2.1.4) of a given site (2). The Site Feature List Result Set MUST be returned and MUST contain one row for each feature when execution is successful to this point, and all the following conditions are met:

- **@MetadataFlag** has the METADATA_WEB flag set
- **@MetadataFlag** has the METADATA_WEB_FEATURES flag set
- **@WebSiteId** is valid
- **@WebId** is valid

The Site Feature List Result set MUST be returned twice with the same schema, once for the default features for the site collection, and again with the features for the specified site (2).

The Site Feature List Result Set is defined in the Common Result Sets Site Feature List Result Set (section 2.2.5.21).

3.1.5.37.10 Unique Permissions Result Set (1)

The Unique Permissions Result Set contains scope and WSS ACL (section 2.2.4.6) information of the site (2) specified by **@WebId**. The Unique Permissions Result Set MUST be returned when execution is successful to this point, and all the following conditions are met:

- **@MetadataFlags** has the METADATA_WEB flag set
- **@MetadataFlags** has the METADATA_WEB_NAVSTRUCT flag set
- The specified site (2) has cached navigation scope information.

The Unique Permissions Result Set is defined in the Common Result Sets Unique Permissions Result Set (section 2.2.5.25).

3.1.5.37.11 NULL Unique Permissions Result Set (1)

The NULL Unique Permissions Result Set MUST be returned when execution is successful to this point, and all the following conditions are met:

- **@MetadataFlags** has the METADATA_WEB flag set
- **@MetadataFlags** has the METADATA_WEB_NAVSTRUCT flag set
- The site (2) specified by **@WebId** or its Parent Site's cached Nav Scope is over 900 characters long.

The NULL Unique Permissions Result Set is defined in the Common Result Sets NULL Unique Permissions Result Set (section 2.2.5.15).

3.1.5.37.12 Empty Result Set

An Empty Result Set containing no rows MUST be returned when execution is successful to this point, and all the following conditions are met:

Copyright © 2014 Microsoft Corporation.

- **@MetadataFlags** has the METADATA_WEB flag set
- **@MetadataFlags** has the METADATA_WEB_NAVSTRUCT flag set
- The site (2) specified by **@WebId** has no cached navigation scope information, or the site is marked as dirty.

The EmptyResult Set is defined in the Common Result Sets Empty Result Set (section 2.2.5.8).

3.1.5.37.13 Redirect Url Result Set

The Redirect Url Result Set contains redirect URL information generated from the **@Url** parameter. If execution is successful to this point, and **@MetadataFlags** has the METADATA_WEB flag set and the METADATA_WEB_WELCOMEPAGE flag set, and the site location specified by **@Url** has a Redirect Type of 255 (None), then **proc_GetTpWebMetaDataAndListMetaData** MUST NOT return this or any further result sets. Otherwise, the Redirect Url Result Set MUST be returned when execution is successful to this point, and all the following conditions are met:

- **@MetadataFlags** has the METADATA_WEB flag set
- **@MetadataFlags** has the METADATA_WEB_WELCOMEPAGE flag set,
- The site location has been marked as a welcome page.

The Redirect Url Result Set MUST contain one row with the Redirect URL and related information for the specified site location.

```
{RedirectType}                tinyint,
{RedirectUrl}                 nvarchar(260),
WelcomePageParameters         ntext;
```

{RedirectType}: Type of item this URL is redirected to. See the Redirect Type section for all possible values for this column**:**

{RedirectUrl}: The complete redirect URL generated from the provided **@Url** parameter.

WelcomePageParameters: The parameters for the welcome page found with this redirect URL. This column can contain a query string starting with "?" or a hash parameter starting with "#".

3.1.5.37.14 No Welcome Redirect Url Result Set

The No Welcome Redirect Url Result Set contains redirect URL information generated from the **@Url** parameter. The No Welcome Redirect Url Result Set MUST be returned when execution is successful to this point, and all the following conditions are met:

- **@MetadataFlags** has the METADATA_WEB flag set
- **@MetadataFlags** has the METADATA_WEB_WELCOMEPAGE flag set,
- The site location specified by @Url has not been marked as a welcome page, that is, it has a RedirectType that is not 0.

The No Welcome Redirect Url Result Set will return exactly one row with the Redirect URL for this folder.

```
{RedirectType}                tinyint,
```


Copyright © 2014 Microsoft Corporation.

```
{RedirectUrl}                   nvarchar(260),
{WelcomePageParameters}         ntext;
```

{RedirectType}: Type of item this URL is redirected to. See the Redirect Type section for all possible values for this column**:**

{RedirectUrl}: The complete redirect URL generated from the provided **@Url** parameter.

{WelcomePageParameters}: This MUST be NULL and can be ignored.

3.1.5.37.15 List Identifier Result Set

The List Identifier Result Set contains identifying information for the file, list (1) or list item specified by **@Url**. If execution is successful to this point, and **@ListId** is NULL, **@MetadataFlags** has the METADATA_FIGURE_TYPE flag set, and the site (2) containing **@Url** is not the site specified by **@WebId**, then **proc_GetTpWebMetaDataAndListMetaData** MUST NOT return this or any further result sets. Otherwise, the List Identifier Result Set MUST be returned when execution is successful to this point, and all the following conditions are met:

- **@ListId is NULL and**
- **@MetadataFlags has the METADATA_FIGURE_TYPE flag set and**
- **@WebId is** a valid site (2) and contains the file, list (1) or list item specified by **@**Url.

```
{ListIdSelected}                uniqueidentifier,
{TypeSelected}                  int,
{ItemIdSelected}                int;
```

{ListIdSelected}: The List Identifier (section 2.2.1.5) of the specified list (1), or of the list (1) containing the specified list item. This value MUST be NULL if **@Url** does not specify a list (1) or list item.

{TypeSelected}: Item type value for the specified list (1). The value of TypeSelected MUST be one of the RedirectType values specified in the Redirect Type section.

{ItemIdSelected}: Document Library Row Identifier of the list item taken from the Docs View (section 2.2.7.4) for the list (1). This value MUST be NULL if **@Url** does not specify a list item.

3.1.5.37.16 List Metadata Result Set (1)

The List Metadata Result Set contains the metadata associated with the list (1) specified by **@ListId** or if **@ListId** is NULL, the containing List specified by **@Url**. The List Metadata Result Set MUST be returned when execution is successful to this point, and **@ListId** specifies a list (1) or **@Url** specifies a list (1) or list item, and **@MetadataFlags** does NOT have the METADATA_LISTMETADATA_NOFETCH flag set. The List Metadata Result Set MUST contain one row if **@WebId** specifies a valid site (2) containing the specified list (1).

The List Metadata Result Set is defined in the Common Result Sets List Metadata Result Set (section 2.2.5.12).

Copyright © 2014 Microsoft Corporation.

3.1.5.37.17 Unique Permissions Result Set (2)

The Unique Permissions Result Set contains all scope and WSS ACL (section 2.2.4.6) information from the specified list (1). The Unique Permissions Result Set MUST be returned when execution is successful to this point, the List Metadata Result Set has been returned, and all the following conditions are met:

- **@MetadataFlags has the METADATA_PREFETCH_SCOPES flag set and**
- **@MetadataFlags** has the METADATA_LISTMETADATA_NOFETCH flag not set.

The Unique Permissions Result Set is defined in Unique Permissions Result Set (section 2.2.5.25).

3.1.5.37.18 NULL Unique Permissions Result Set (2)

The NULL Unique Permissions Result Set MUST be returned when execution is successful to this point, the List Metadata Result Set has been returned, and all the following conditions are met:

- **@MetadataFlags has the METADATA_PREFETCH_SCOPES flag set**
- **@MetadataFlags has the METADATA_LISTMETADATA_NOFETCH flag not set.**
- **Permissions** associated with the specified list (1) do not exist.

The NULL Unique Permissions Result Set is defined by the NULL Unique Permissions Result Set (section 2.2.5.15).

3.1.5.37.19 Event Receivers Result Set (2)

The Event Receivers Result Set contains information about the event receivers defined for the site (2) containing the list (1) specified by **@ListId**, or if **@ListId** is NULL, the containing list (1) specified by **@Url**. The Event Receivers Result Set MUST be returned when execution is successful to this point, and the List Metadata Result Set has been returned.

The Event Receivers Result Set MUST contain one row for each event receiver with an Event Host Type (section 2.2.3.5) of 2 (List) registered for the site specified by **@WebId**.

The Event Receivers Result Set is defined in the Common Result Sets Event Receivers Result Set (section 2.2.5.9).

3.1.5.37.20 List Web Parts Result Set (1)

The List Web Parts Result Set contains information about the List web parts defined for the list (1) specified by **@ListId**, or if **@ListId** is NULL, the containing list (1) specified by **@Url**. The List Web Parts Result Set MUST be returned when execution is successful to this point, the List Metadata Result Set has been returned, and **@MetadataFlags** has the METADATA_FETCH_VIEWS flag set.

The List Web Parts Result Set MUST contain one row for each WebPart registered for this List.

The List Web Parts Result Set is defined in the Common Result Sets List Web Parts Result Set (section 2.2.5.13).

Copyright © 2014 Microsoft Corporation.

3.1.5.37.21 List Metadata Result Set (2)

The List Metadata Result Set contains the metadata associated with a specified User Info List. The List Metadata Result Set MUST be returned when execution is successful to this point, and all the following conditions are met:

- **@WebSiteId** contains a valid Site Collection Identifier (section 2.2.1.9) and the site collection has a User Info List,
- The List Identifier (section 2.2.1.5) specified by **@ListId**, or if **@ListId** is NULL, the List Identifier of the list (1) at (or that contains) the location specified by **@Url** doesn't match the List Identifier for the User Info List,
- **@MetadataFlags** has the METADATA_USERINFOLIST flag set, or @Url specifies a list (1) or a location within a list (1), and
- **@MetadataFlags** doesn't have the METADATA_USERINFOLIST_NOFETCH flag set.

The List Metadata Result Set MUST contain one row if it is returned.

The List Metadata Result Set is defined in the Common Result Sets List Metadata Result Set (section 2.2.5.12).

3.1.5.37.22 Unique Permissions Result Set (3)

The Unique Permissions Result Set contains all scope and WSS ACL (section 2.2.4.6) information from a specified User Info List. The Unique Permissions Result Set MUST be returned when execution is successful to this point, and all the following conditions are met:

- **@WebSiteId** contains a valid Site Collection Identifier (section 2.2.1.9) and the site collection has a User Info List,
- The List Identifier (section 2.2.1.5) specified by **@ListId**, or if **@ListId** is NULL, the List Identifier of the list (1) at (or that contains) the location specified by **@Url** doesn’t match the List Identifier for the User Info List,
- **@MetadataFlags** has the METADATA_USERINFOLIST flag set, or **@Url** specifies a list (1) or a location within a list (1), and
- **@MetadataFlags** doesn't have the METADATA_USERINFOLIST_NOFETCH flag set
- **@MetadataFlags** has the METADATA_PREFETCH_SCOPES flag set

The Unique Permissions Result Set is defined in Unique Permissions Result Set (section 2.2.5.25) section.

3.1.5.37.23 NULL Unique Permissions Result Set (3)

The NULL Unique Permissions Result Set MUST be returned when execution is successful to this point, and all the following conditions are met:

- **@WebSiteId** contains a valid Site Collection Identifier (section 2.2.1.9) and the site collection has a User Info List
- The List Identifier (section 2.2.1.5) specified by **@ListId**, or if **@ListId** is NULL, the List Identifier of the list (1) at (or that contains) the location specified by **@Url** doesn’t match the List Identifier for the User Info List

Copyright © 2014 Microsoft Corporation.

- **@MetadataFlags** has the METADATA_USERINFOLIST flag set, or **@Url** specifies a list (1) or a location within a list (1), and
- **@MetadataFlags** doesn't have the METADATA_USERINFOLIST_NOFETCH flag set
- **@MetadataFlag**s has the METADATA_PREFETCH_SCOPES flag set

No unique permissions exist for the root site of the site collection.

The NULL Unique Permissions Result Set is defined in the in the Common Result Sets NULL Unique Permissions Result Set (section 2.2.5.15).

3.1.5.37.24 Event Receivers Result Set (3)

The Event Receivers Result Set contains information about the event receivers defined for the site containing the specified User Info List. The Event Receivers Result Set MUST be returned when execution is successful to this point, and all the following conditions are met:

- **@WebSiteId** contains a valid Site Collection Identifier (section 2.2.1.9) and the site collection has a User Info List
- The List Identifier (section 2.2.1.5) specified by **@ListId**, or if **@ListId** is NULL, the List Identifier of the list (1) at (or that contains) the location specified by **@Url** doesn't match the List Identifier for the User Info List
- **@MetadataFlags** has the METADATA_USERINFOLIST flag set, or **@Url** specifies a list (1) or a location within a list (1)
- **@MetadataFlags** has the METADATA_USERINFOLIST_FULL flag set
- **@MetadataFlags** doesn't have the METADATA_USERINFOLIST_NOFETCH flag set

The Event Receivers Result Set MUST contain one row for each event receiver with an Event Host Type (section 2.2.3.5) of 2 (List) that is registered for the root site of the site collection specified by **@WebSiteId**.

The Event Receivers Result Set is defined in the Common Result Sets Event Receivers Result Set (section 2.2.5.9).

3.1.5.37.25 List Web Parts Result Set (2)

The List Web Parts Result Set contains information about the List Web Parts defined for the specified User Info List. The List Web Parts Result Set MUST be returned if execution is successful to this point, and all the following conditions are met:

- **@WebSiteId** contains a valid Site Collection Identifier (section 2.2.1.9) and the site collection has a User Info List
- The List Identifier (section 2.2.1.5) specified by **@ListId**, or if **@ListId** is NULL, the List Identifier of the list (1) at (or that contains) the location specified by **@Url** doesn't match the List Identifier for the User Info List
- **@MetadataFlags** has the METADATA_USERINFOLIST flag set, or **@Url** specifies a list (1) or a location within a list (1)
- **@MetadataFlags** has the METADATA_USERINFOLIST_FULL flag set

Copyright © 2014 Microsoft Corporation.

- **@MetadataFlags** doesn't have the METADATA_USERINFOLIST_NOFETCH flag set

The List Web Parts Result Set MUST contain one row for each WebPart registered for this List.

The List Web Parts Result Set is defined in the Common Result Sets List Web Parts Result Set (section 2.2.5.13).

3.1.5.37.26 Document Metadata Result Set

The Document Metadata Result Set contains the metadata for a single document when the provided **@Url** parameter refers to a single document. The Document Metadata Result Set MUST be returned when execution is successful to this point, and either of the following conditions sets are met:

- **@MetadataFlags** has the METADATA_URL flag set
- The document is not contained within a list (1)

  OR

- **@MetadataFlags** has the METADATA_URL_REALLY flag set

The Document Metadata Result Set MUST return one row when **@Url** is valid. The Document Metadata Result Set MUST be ordered by the DocId column, and is defined in the Common Result Sets Document Metadata Result Set (section 2.2.5.6).

3.1.5.37.27 NULL Result Set

The NULL Result Set returns no data. The NULL Result Set MUST be returned if the Document Metadata Result Set is not empty, otherwise the NULL Result Set MUST NOT be returned. The NULL Result Set MUST return zero rows in a schema containing a single unnamed column.

3.1.5.38 proc_GetUniqueScopesInList

The **proc_GetUniqueScopesInList** stored procedure is invoked to get the ACL information for all the unique Security Scopes of folders and list items contained in a specified list (1).

```
PROCEDURE proc_GetUniqueScopesInList(
      @SiteId                       uniqueidentifier,
      @WebId                        uniqueidentifier,
      @ListId                       uniqueidentifier,
      @GetAll                       bit
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the list (1).

@WebId: The Site Identifier (section 2.2.1.11) of the site that contains the list (1).

@ListId: The List Identifier (section 2.2.1.5) of the list (1) for which the ACL information is requested.

@GetAll: A bit specifying whether to fetch ACL information for all unique Security Scopes. If this bit is set to "0", ACL information for at most 1,001 Security Scopes MUST be returned in the Unique Permissions Result Set (section 2.2.5.25), else ACL information for all unique Security Scopes MUST be returned in the Unique Permissions Result Set.

Copyright © 2014 Microsoft Corporation.

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

The **proc_GetUniqueScopesInList** stored procedure MUST return a single result set.

3.1.5.38.1 Unique Permissions Result Set

The Unique Permissions Result Set returns the ACL information for all the unique Security Scopes of Folders and List Items contained in a specified List. The Unique Permissions Result Set MUST be returned if the List has 1 or more Folders or List Items with unique Security Scopes. The Unique Permissions Result Set MUST return one row for every folder or list item with a unique Security Scope. If the **@GetAll** bit is set to "0", this result set MUST return at most 1,001 rows.

The Unique Permissions Result Set is defined the Common Result Sets Unique Permissions Result Set (section 2.2.5.25).

3.1.5.38.2 NULL Unique Permissions Result Set

The NULL Unique Permissions Result Set MUST return one row if the List has no Folders or List Items with unique Security Scopes. The NULL Unique Permissions Result Set is defined in the Common Result Sets NULL Unique Permissions Result Set (section 2.2.5.15).

3.1.5.39 proc_GetVersion

The **proc_GetVersion** stored procedure is invoked to get the component version number string associated with the specified version identifier GUID (using the format major.minor.phase.build; for example, 3.0.106.0).

```
PROCEDURE proc_GetVersion(
      @VersionId                    uniqueidentifier,
      @Version                      nvarchar(64)       OUTPUT
);
```

@VersionId: The Version Identifier of the component version number to retrieve.

@Version: An output parameter containing the component version number string matching the specified version identifier. This value MUST be unchanged from the input value if the specified **@VersionId** value does not have a matching component version number.

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

Value	Description
0	Successful execution.

The **proc_GetVersion** stored procedure MUST NOT return any result sets.

The client can define one or more version numbers. Each of the version number is identified by a Version Id. When a database is created by the client on the back-end database server, the client stores the version numbers with different version identifiers in the Versions table (section 2.2.7.11) in the back-end database server. When the client connects to a back-end database server, the client SHOULD retrieve the Version Ids using the stored procedure **proc_GetVersion**<12> to make sure the versions are within an acceptable range defined by the client; otherwise the client MUST refuse to connect to the backend database server.

Copyright © 2014 Microsoft Corporation.

The version identifiers and the acceptable version numbers for this protocol used when communicating with a content database are listed in the following table.

Version Id GUID	Acceptable version range
'00000000-0000-0000-0000-000000000000'	12.0.6425.1000 – 12.0.6425.1000
'6333368D-85F0-4EF5-8241-5252B12B2E50'	3.1.8.0 – 3.1.8.0

When the protocol is used to communicate with a configuration database, the list of Version Ids used are within the following table:

Version Id GUID	Acceptable version range
'00000000-0000-0000-0000-000000000000'	12.0.6425.1000 – 12.0.6425.1000
'F4D348C4-A6E9-4ed5-BDB2-2358B74EF902'	3.0.9.0 – 3.0.9.0

The acceptable version numbers specified in [table A] and [table B] MAY change when the client is updated through a service pack. As a result, the updated client might not be able to communicate with the back-end database server any more. In order to re-enable the client to server communication, the client MUST use an upgrade process to modify any data structure on the back-end database server to accommodate any changes that might have occurred to this protocol in the service pack, and update the version numbers to match the new acceptable version numbers. The upgrade logic is application implementation specific.

3.1.5.40 proc_GetWebMetaInfo

The **proc_GetWebMetaInfo** stored procedure is invoked to request metadata information for a site.

```
PROCEDURE proc_GetWebMetaInfo(
      @WebSiteId                    uniqueidentifier,
      @WebDirName                   nvarchar(256),
      @WebLeafName                  nvarchar(128),
      @DGCacheVersion               bigint,
      @SystemId                     varbinary(512) = NULL
);
```

@WebSiteId: The Site Collection Identifier (section 2.2.1.9) for the site collection containing the Site whose metadata is requested.

@WebDirName: The directory name part of the Site location.

@WebLeafName: The leaf name part of the Site location.

@DGCacheVersion: The version number of the Domain Group Map Cache in the WFE. This can be compared with the Domain Group Map Cache version number on the BEDS returned in the Domain Group Cache Versions Result Set to determine whether updates are needed. A **@DGCacheVersion** value of -2 specifies that information about the Domain Group Map Cache is not requested, and the Domain Group Cache BEDS Update Result Set (section 2.2.5.3) and the Domain Group Cache WFE Update Result Set (section 2.2.5.5) MUST NOT be returned.

@SystemId: The **SystemID** (section 2.2.1.12) of the current user requesting the Site metadata information.

Copyright © 2014 Microsoft Corporation.

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

Value	Description
0	Successful execution.
3	A Site was not found at the specified location.
1271	The Site is locked.

The **proc_GetWebMetaInfo** stored procedure MUST return two, five, or six result sets as follows:

3.1.5.40.1 Site Categories Result Set

The Site Categories Result Set contains information about categories associated with the specified site (2).

The Site Categories Result Set MUST be returned and MUST contain one row for each category specified for the site (2). If the specified site (2) is not found, zero rows MUST be returned.

The Site Categories Result Set is defined in the Common Result Sets Site Categories Result Set (section 2.2.5.22).

3.1.5.40.2 Site MetaInfo Result Set

The Site MetaInfo Result Set MUST be returned. The Site MetaInfo Result Set MUST contain a single row containing the Metainfo for a valid site (2). If the specified site is not found, zero rows MUST be returned and the stored procedure MUST NOT return any more result sets. The Site MetaInfo Result Set is defined in the Common Result Sets Site MetaInfo Result Set (section 2.2.5.24).

3.1.5.40.3 Domain Group Cache Versions Result Set

The Domain Group Cache Versions Result Set contains information about the version numbers associated with the Domain Group Map Caches on the WFE and on the BEDS for the specified site collection.

The Domain Group Cache Versions Result Set MUST be returned if the specified site (2) is found, and MUST contain one row of version number data. If the specified **@DGCacheVersion** value is -2, then all columns returned MUST have the value -2, indicating that the value MUST NOT be used for comparison. The Domain Group Cache Versions Result Set is defined in the Common Result Sets Domain Group Cache Versions Result Set (section 2.2.5.4).

3.1.5.40.4 Domain Group Cache BEDS Update Result Set

The Domain Group Cache BEDS Update Result Set contains information to be used in recomputing the domain group cache.

The Domain Group Cache BEDS Update Result Set returns only if the Domain Group Cache Versions Result Set is returned and **@DGCacheVersion** is not -2 ('Skip') and the real domain group version 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 BEDS Update Result Set is returned, it indicates that the database's copy of the Domain Group Map Cache is out of date and MUST be recomputed to ensure that proper security checks can be made.

Copyright © 2014 Microsoft Corporation.

When returned, the Domain Group Cache BEDS Update Result Set MUST contain one row for each domain group that is a member of a site group in the site collection, ordered by the User Identifier (section 2.2.1.13) of the domain groups.

The result set is defined in the Common Result Sets Domain Group Cache BEDS Update Result Set (section 2.2.5.3).

3.1.5.40.5 Domain Group Cache WFE Update Result Set

The Domain Group Cache WFE Update Result Set contains the binary data needed to refresh the domain group map cache.

The Domain Group Cache WFE Update Result Set returns only if the Domain Group Cache Versions Result Set is returned and **@DGCacheVersion** is not -2 ('Skip') 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).

If the Domain Group Cache WFE Update Result Set is returned, one row MUST be returned. The Domain Group Cache WFE Update Result Set is defined in the Common Result Sets Domain Group Cache WFE Update Result Set (section 2.2.5.5).

3.1.5.40.6 Site Metadata Result Set

The Site Metadata Result Set contains metadata for the specified site (2).

The Site Metadata Result Set MUST be returned if the specified site (2) is found, and MUST contain a single row corresponding to the specified Site. If the site (2) is locked, the stored procedure MUST NOT return any more result sets.

The Site Metadata Result Set is defined in the Common Result Sets Site Metadata Result Set (section 2.2.5.23).

3.1.5.40.7 Event Receivers Result Set

The Event Receivers Result Set contains information about the event receivers defined for the specified site (2). There MUST be one row in the Event Receivers Result Set for each event receiver that is registered for this site. The Event Receivers Result Set MUST be returned on successful execution.

The Event Receivers Result Set is defined in the Common Result Sets Event Receivers Result Set (section 2.2.5.9).

3.1.5.41 proc_GetWebMetainfoByUrl

The **proc_GetWebMetainfoByUrl** stored procedure is invoked to request site metadata information for the Site containing a specified URL.

```
PROCEDURE proc_GetWebMetainfoByUrl(
      @WebSiteId                    uniqueidentifier,
      @Url                          nvarchar(260),
      @DGCacheVersion               bigint,
      @SystemId                     varbinary(512)     = NULL
);
```


Copyright © 2014 Microsoft Corporation.

@WebSiteId: The Site Collection Identifier (section 2.2.1.9) for the site collection containing the site (2) or document whose metadata is requested.

@Url: A complete store-relative form URL for the Site or for a document within the Site.

@DGCacheVersion: The version number of the Domain Group Map Cache in the WFE. This can be compared with the Domain Group Map Cache version number on the BEDS returned in the Domain Group Cache Versions Result Set to determine whether updates are needed. A **@DGCacheVersion** value of -2 specifies that information about the Domain Group Map Cache is not requested, and the Domain Group Cache BEDS Update Result Set (section 2.2.5.3) and the Domain Group Cache WFE Update Result Set (section 2.2.5.5) MUST NOT be returned.

@SystemId: The **SystemID** (section 2.2.1.12) of the current user requesting the Site Metadata information.

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

Value	Description
0	Successful execution.
1168	The site collection does not exist.
1271	The Site is locked.

The **proc_GetWebMetainfoByUrl** stored procedure MUST return zero, five, six, or seven result sets. Some of the result sets are returned conditionally, and all result sets returned will be sent in the order specified in the following.

3.1.5.41.1 Site URL Result Set

The Site URL Result Set MUST be returned if the values in **@Url** and **@WebSiteId** are valid. Otherwise, the stored procedure MUST NOT return any result sets. The Site URL Result Set MUST contain a single row, with the store-relative form URL of the site (2) containing the URL specified in the **@Url** parameter.

```
{WebUrl}                    nvarchar(260);
```

{WebUrl}: The store-relative form URL of the Site that contains the URL specified in the **@Url** parameter.

3.1.5.41.2 Site Categories Result Set

The Site Categories Result Set lists the unordered set of categories defined for content in this site (2), one category per row. If no categories are defined for the site (2), the Site Categories Result Set MUST contain no rows. The Site Categories Result Set MUST be returned on successful determination of the site (2).

The Site Categories Result Set is defined in the Common Result Sets Site Categories Result Set (section 2.2.5.22).

Copyright © 2014 Microsoft Corporation.

3.1.5.41.3 Site Metainfo Result Set

The Site MetaInfo Result Set MUST be returned on successful determination of the site (2). The Site MetaInfo Result Set MUST contain a single row containing the Metainfo for the site (2).

The Site Metainfo Result Set is defined in the Common Result Sets Site MetaInfo Result Set (section 2.2.5.24).

3.1.5.41.4 Domain Group Cache Versions Result Set

The Domain Group Cache Versions Result Set contains information about the version numbers associated with the Domain Group Map Caches on the WFE and on the BEDS for the specified site collection.

The Domain Group Cache Versions Result Set MUST be returned on successful determination of the site (2) and MUST contain one row of version number data. If the specified **@DGCacheVersion** value is -2, then all columns returned MUST have the value -2, indicating that the value MUST NOT be used for comparison. The Domain Group Cache Versions Result Set is defined in the Common Result Sets Domain Group Cache Versions Result Set (section 2.2.5.4).

3.1.5.41.5 Domain Group Cache BEDS Update Result Set

The Domain Group Cache BEDS Update Result Set contains information to be used in re-computing the domain group map cache, which contains the mapping of domain groups to the site groups they are members of. The presence of the Domain Group Cache BEDS Update Result Set means the database's copy of the domain group map cache is out of date, and MUST be recomputed to ensure that proper security checks can be made.

The Domain Group Cache BEDS UpdateResult Set MUST be returned on successful determination of the site (2) if **@DGCacheVersion** does not equal "-2" and the value of RealVersion is greater than the value of CachedVersion in the results of the Domain Group Cache Versions Result Set. Otherwise, this result set MUST NOT be returned.

If the Domain Group Cache BEDS Update Result Set is returned, it MUST contain one row for each domain group which is a member of a site group in the site collection, ordered by the User Identifier (section 2.2.1.13) of the domain groups.

The Domain Group Cache BEDS Update Result Set is defined in the Common Result Sets Domain Group Cache BEDS Update Result Set (section 2.2.5.3).

3.1.5.41.6 Domain Group Cache WFE Update Result Set

The Domain Group Cache WFE Update Result Set contains the binary data needed to refresh the domain group map cache. If the Domain Group Cache WFE Update Result Set is returned, it indicates that the BEDS Domain Group map cache is up to date, and the WFE cache can be refreshed if necessary.

The Domain Group Cache WFE Update Result Set MUST be returned on successful determination of the site (2) if **@DGCacheVersion** does not equal "-2" and the value of RealVersion is less than or equal to the value of CachedVersion in the results of the Domain Group Cache Versions Result Set. Otherwise, the Domain Group Cache WFE Update Result Set MUST NOT be returned.

If the Domain Group Cache WFE Update Result Set is returned, it MUST contain one row.

The Domain Group Cache WFE Update Result Set is defined in the Common Result Sets Domain Group Cache WFE Update Result Set (section 2.2.5.5).

Copyright © 2014 Microsoft Corporation.

3.1.5.41.7 Site Metadata Result Set

The Site Metadata Result Set contains specialized Site metadata. The Site Metadata Result Set MUST be returned on successful determination of the site (2). If the site is locked, the stored procedure MUST NOT return any more result sets. There MUST be one row in the Site Metadata Result Set.

The Site Metadata Result Set is defined in the Common Result Sets Site Metadata Result Set (section 2.2.5.23).

3.1.5.41.8 Event Receivers Result Set

The Event Receivers Result Set contains information about the event receivers defined for this site (2). There MUST be one row in the Event Receivers Result Set for each event receiver registered for this site (2). The Event ReceiversResult Set MUST be returned on successful completion.

The Event Receivers Result Set is defined in the Common Result Sets Event Receivers Result Set (section 2.2.5.9).

3.1.5.42 proc_ListDocumentVersions

The **proc_ListDocumentVersions** stored procedure is invoked to list all available version history information for a specified document.

```
PROCEDURE proc_ListDocumentVersions(
      @DocSiteId                    uniqueidentifier,
      @DocWebId                     uniqueidentifier,
      @DocDirName                   nvarchar(256),
      @DocLeafName                  nvarchar(128),
      @MaxLevel                     tinyint,
      @UserId                       int
);
```

@DocSiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the document.

@DocWebId: The Site Identifier (section 2.2.1.11) of the site (2) containing the document.

@DocDirName: The folder name containing the document.

@DocLeafName: The leaf name containing the document.

@MaxLevel: A Publishing Level Type (section 2.2.2.6) value indicating the maximum publishing level value of the document that MUST be returned to the WFE in the Document Versions Result Set (section 3.1.5.42.3) if multiple publishing levels of the document are available. See **@UserId** for an exception to this.

@UserId: The User Identifier (section 2.2.1.13) of the current user, used for access rights verification. If the current user is the owner of one or more versions of the document at any publishing level, the version information MUST be returned in the Document Versions Result Set, ignoring the value specified in **@**MaxLevel.

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

Copyright © 2014 Microsoft Corporation.

Value	Description
0	Successful execution.
2	The Document specified is not a file (that is, it does not have a Document Store Type (section 2.2.2.4) of 0).

The **proc_ListDocumentVersions** stored procedure MUST return two or three result sets upon successful completion.

3.1.5.42.1 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 document URL is within a list (1) or document library, security information is returned from the effective scope for the document URL.

The Individual URL Security Result Set MUST only be returned if the document URL is contained within a List or document library. Otherwise, the NULL Individual URL Security Result Set (section 2.2.5.14) MUST be returned instead. If returned, the Individual URL Security Result Set MUST contain a single row.

The Individual URL Security Result Set is defined in the Common Result Sets Individual URL Security Result Set (section 2.2.5.10).

3.1.5.42.2 NULL Individual URL Security Result Set

The NULL Individual URL Security Result Set indicates that the specified document URL is not contained within a List or document library. The NULL Individual URL Security Result Set MUST only be returned if the document URL is not contained within a List or document library.

The NULL Individual URL Security Result Set MUST return a single row and is defined in the Common Result Sets NULL Individual URL Security Result Set (section 2.2.5.14).

3.1.5.42.3 Document Versions Result Set

The Document Versions Result Set returns version information for a specified document. The Document Versions Result Set MUST only be returned when the specified document has a Document Store Type (section 2.2.2.4) of 0, indicating that it is a file. The Document Versions Result Set MUST contain document version information corresponding to publishing levels less than or equal to the specified publishing level, or corresponding to any publishing levels owned by the current user.

```
TimeCreated                 datetime,
Version                     int,
Size                        int,
CheckinComment              nvarchar(1023),
MetaInfo                    image,
{IsTip}                     bit,
Level                       tinyint,
DraftOwnerId                int;
```

TimeCreated: A timestamp in UTC format specifying when this document was created.

Version: A counter incremented any time a change is made to this document, used for internal conflict detection.

Copyright © 2014 Microsoft Corporation.

Size: The number of bytes in the document stream.

CheckinComment: An optional user-provided description used when a document is being checked in or published. This value MUST be NULL for Documents that have not been checked in or published.

MetaInfo: A metadict for the document. The metadict format is specified in [MS-FPSE] section 2.2.2.2.11.

{IsTip}: If set to 1, the document is a current version, otherwise it is a historical version.

Level: A Publishing Level Type (section 2.2.2.6) value specifying the publishing status of the document.

DraftOwnerId: The User that published this document as a draft. This value is only non-NULL if the document is a draft version.

3.1.5.43 proc_ListUrls

The **proc_ListUrls** stored procedure is invoked to get metadata for a document specified by a URL and the Documents contained within it, if any.

```
PROCEDURE proc_ListUrls(
      @DirSiteId                    uniqueidentifier,
      @DirWebId                     uniqueidentifier,
      @DirFullUrl                   nvarchar(260),
      @AttachmentsFlag              tinyint,
      @ClientTimeStamp              datetime,
      @FetchLinkInfo                bit,
      @IncludeThicketDirs           bit,
      @IncludeListItems             bit,
      @UserId                       int
);
```

@DirSiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the document specified by a URL.

@DirWebId: The Site Identifier (section 2.2.1.11) of the site (2) containing the document.

@DirFullUrl: The store-relative form URL specifying the document.

@AttachmentsFlag: An Attachments Flag (section 2.2.3.1) value specifying whether the document is, or is contained within, a folder for attachments.

@ClientTimeStamp: A datetime that specifies a limiting time on the data in the result sets returned. See the description of the result sets below for the specific effects of this parameter value.

@FetchLinkInfo: A bit flag specifying whether to return the Link Info Result Set (section 3.1.5.26.5). If this parameter is set to 1 and the specified document is a folder, the Link Info Result Set MUST be returned.

@IncludeThicketDirs: A bit flag specifying whether to return **thicket folders** in the Contained Document Metadata Result Set (section 3.1.5.43.7). If this parameter is set to 1 and the specified document is a folder, any thicket folders MUST be included in the Contained Document Metadata Result Set.

Copyright © 2014 Microsoft Corporation.

@IncludeListItems: A bit flag specifying whether to include list items which are files in the Link Info Result Set and the Contained Document Metadata Result Set. If this parameter is set to 1 and the document is a folder, List Items with a Document Store Type (section 2.2.2.4) of 0 (File) MUST be included in the Link Info Result Set and the Contained Document Metadata Result Set.

@UserId: The User Identifier (section 2.2.1.13) for the current user.

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

Value	Description
0	Successful execution.
3	The document URL is not valid, or the document is not a folder or Site.

The **proc_ListUrls** stored procedure MUST return two to six result sets, as described below, in the following order:

3.1.5.43.1 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 document URL is within a list (1) or document library, security information is returned from the effective scope for the document URL.

The Individual URL Security Result Set MUST only be returned if the document URL is contained within a List or document library. Otherwise, the NULL Individual URL Security Result Set (section 3.1.5.43.2) MUST be returned instead. If returned, the Individual URL Security Result Set MUST contain a single row.

The Individual URL Security Result Set is defined in the Common Result Sets Individual URL Security Result Set (section 2.2.5.10).

3.1.5.43.2 NULL Individual URL Security Result Set

The NULL Individual URL Security Result Set indicates that the specified document URL is not contained within a List or document library. The NULL Individual URL Security Result Set MUST only be returned if the document URL is not contained within a List or document library.

The NULL Individual URL Security Result Set MUST return a single row, and is defined in the Common Result Sets NULL Individual URL Security Result Set (section 2.2.5.14).

3.1.5.43.3 Server Time Result Set

The Server Time Result Set returns the current time from the BEDS in UTC. The Server Time Result Set MUST be returned and MUST contain a single row of data.

The Server Time Result Set is defined in the Common Result Sets Server Time Result Set (section 2.2.5.18).

3.1.5.43.4 Subsite List Result Set

The Subsite List Result Set contains an unordered list of store-relative form URLs for all subsites whose parent site (2) is the site specified by the **@DirWebId** parameter.

Copyright © 2014 Microsoft Corporation.

If the specified document is not a folder, the Subsite List Result Set MUST NOT be returned. Otherwise the Subsite List Result Set MUST be returned and MUST contain one row for each subsite within the specified site (2), and it MUST contain no rows if there are no such subsites.

The Subsite List Result Set is defined the URL Result Set (section 2.2.5.26).

3.1.5.43.5 Document Metadata Result Set

The Document Metadata Result Set contains the metadata for the specified document. If the document is not a folder, or if the **@DirFullUrl** parameter contains an empty string, the Document Metadata Result Set MUST NOT be returned. Otherwise, the Document Metadata Result Set MUST contain one row with the metadata information for the document.

```
Id                          uniqueidentifier,
{FullUrl}                   nvarchar(260),
Type                        tinyint,
MetaInfoTimeLastModified    datetime,
MetaInfo                    image,
Size                        int,
TimeCreated                 datetime,
TimeLastModified            datetime,
Version                     int,
DocFlags                    int,
{ListType}                  int,
tp_Name                     nvarchar(38),
{ListTitle}                 nvarchar(255),
CacheParseId                uniqueidentifier,
{GhostDirName}              nvarchar(256),
{GhostLeafName}             nvarchar(128),
{tp_Login}                  nvarchar(255),
{CheckoutDate}              datetime,
{CheckoutExpires}           datetime,
VirusStatus                 int,
VirusInfo                   nvarchar(255),
SetupPathVersion            tinyint,
SetupPath                   nvarchar(255),
SetupPathUser               nvarchar(255),
NextToLastTimeModified      datetime,
UIVersion                   int,
CheckinComment              nvarchar(1023),
WelcomePageUrl              nvarchar(260),
WelcomePageParameters       ntext,
tp_Flags                    bigint,
Acl                         image,
AnonymousPermMask           bigint,
DraftOwnerId                int,
Level                       tinyint,
ParentVersion               int,
TransformerId               uniqueidentifier,
ParentLeafName              nvarchar(128),
ProgId                      nvarchar(255),
DoclibRowId                 int,
tp_DefaultWorkflowId        uniqueidentifier,
ListId                      uniqueidentifier;
```

Id: The Document Identifier (section 2.2.1.2) of the document.

Copyright © 2014 Microsoft Corporation.

{FullUrl}: The store-relative form URL for the document.

Type: The Document Store Type (section 2.2.2.4) of the document.

MetaInfoTimeLastModified: A datetime with a timestamp in UTC format specifying the last time the Metainfo value of the document was changed. This value can be NULL.

MetaInfo: A metadict for the document. The metadict format is specified in [MS-FPSE] section 2.2.2.2.11. This value can be NULL.

Size**:** The number of bytes in the document stream of the document. This value can be NULL.

TimeCreated: A datetime with a timestamp in UTC format specifying when the document was created.

TimeLastModified: A datetime with a timestamp in UTC format specifying when the document was last modified.

Version: A counter incremented any time a change is made to the document, and used for internal conflict detection.

DocFlags: A Doc Flags (section 2.2.2.3) value describing the document. This value can be NULL.

{ListType}: A packed combination of the List Base Type (section 2.2.3.11) and List Server Template (section 2.2.3.12) values of the List containing this document, consisting of the List Server Template value multiplied by 256 and added to the value of the List Base Type.

tp_Name: The List Identifier (section 2.2.1.5) of the List containing this document.

{ListTitle}: If the document URL is the root folder of a List, this contains the display name of the List. Otherwise, this value MUST be NULL.

CacheParseId: This value MUST be NULL.

{GhostDirName}: This value MUST be NULL.

{GhostLeafName}: This value MUST be NULL.

{tp_Login}: This value MUST be NULL.

{CheckoutDate}: This value MUST be NULL.

{CheckoutExpires}: This value MUST be NULL.

VirusStatus: A Virus Status (section 2.2.3.18) value specifying the current virus state of the document. This value can be NULL if the document has not been processed by a virus scanner.

VirusInfo: A string containing a provider-specific message returned by the virus scanner when it last processed the document. This value can be NULL if the document has not been processed by a virus scanner.

SetupPathVersion: If this is a ghosted document, this specifies the setup path location that the SetupPath fragment is relative to. This value MUST NOT be NULL. The following are valid values.

Value	Description
2	The SetupPath is relative to the install location of Windows SharePoint Services 2.0 on the WFE (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\60)

Copyright © 2014 Microsoft Corporation.

Value	Description
3	The SetupPath is relative to the install location of Windows SharePoint Services 3.0 on the WFE (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\12).

SetupPath: If the document is now or once was ghosted, this contains the setup path fragment relative to the base setup path described above by the **SetupPathVersion** value, where the content stream of the document can be found. This value can be NULL.

SetupPathUser: If the document is now or once was ghosted, this contains the login name of the user who created the ghosted document. This value can be NULL.

NextToLastTimeModified: The value of **TimeLastModified** from the previous time when the document was last saved. If the **NextToLastTimeModified** value matches the **TimeLastModified** value when the change occurred and the client has a version of the document that it has successfully modified, the client can safely submit the document to the server despite what appears to be an intervening edit to the document. This value can be NULL.

UIVersion: The UI version number for the document.

CheckinComment: An optional user-supplied description provided when the document is checked in or published. This value can be NULL.

WelcomePageUrl: If the document is a folder, this specifies an optional page to redirect to when the folder is requested with an HTTP GET operation. The URL is relative to the URL of the folder itself, and MUST be contained within that folder. Attempts to break out of the folder, such as "../../somepage.aspx", are not valid. This value can be NULL.

WelcomePageParameters: Contains optional "search" parameters to specify as part of the **WelcomePageUrl** value. These parameters start at either the query string signifier '?' or the bookmark signifier '#'. This value can be NULL.

tp_Flags: The List Flags (section 2.2.2.5) value for the List that contains the document.

Acl: The binary serialization of the WSS ACL (section 2.2.4.6) for the document. The WSS ACL is either explicitly defined for the document, or inherited from the parent object of the document. This value can be NULL.

AnonymousPermMask: A WSS Rights Mask that indicates the rights granted to a user who is anonymous, or has no specific rights, to the document. This value can be NULL.

DraftOwnerId: The User Identifier (section 2.2.1.13) of the User that published the document as a draft. This value MUST be NULL if the document is not a draft version.

Level: A Publishing Level Type (section 2.2.2.6) value specifying the publishing status of the document.

ParentVersion: If the document is a transformed version of another document, this is the UI version value from the original document. This value MUST be NULL if the document is not the product of a document transformation.

TransformerId: If the document is a transformed version of another document, this is the Globally Unique Identifier (GUID) of the agent that performed the transformation. This value MUST be NULL if the document is not the product of a document transformation.

ParentLeafName: If the document is a transformed version of another document, this is the leaf name of the original document. The original document MUST be in the same folder as the

Copyright © 2014 Microsoft Corporation.

transformed document. If either is moved, the relationship is broken. This value MUST be NULL if the document is not the product of a document transformation.

ProgId: Specifies a preferred application to open the document. The ProgId value is used to distinguish between different applications that save files with a given file extension (for example, different editors for .HTML or .XML files). This value MUST be NULL if a ProgId was not specified when the document was saved.

DoclibRowId: The document library row identifier for the document. If the document is not contained in a List, this value MUST be NULL.

tp_DefaultWorkflowId: The Workflow Identifier (section 2.2.1.16) corresponding to the workflow (2) to be invoked if the document is in a moderated List and the document is submitted for approval as part of a check in.

ListId: The List Identifier of the List that contains the document. If the document is not contained in a List, this value MUST be NULL.

3.1.5.43.6 Link Info Result Set

The Link Info Result Set returns information about all forward links from and backward links to the Documents contained within the specified document. If the specified document is not a folder, or the **@FetchLinkInfo** parameter is not set to 1, the Link Info Result Set MUST NOT be returned. Otherwise, if the document is a folder and the **@FetchLinkInfo** parameter is set to 1, the Link Info Result Set MUST be returned. The Link Info Result Set MUST contain one row for each Link that has been modified after the value in the **@ClientTimeStamp** parameter for each contained document in the site collection which has its directory name equal to the value of the **@DirFullUrl** parameter, and the contained document has a Document Store Type (section 2.2.2.4) of 0 (File) and the **@IncludeListItems** parameter is set to 1, the document is in a document library, or the document can have a document stream, and for forward links, is a published or draft version which is not checked out to the specified User, or is a checked out version which is checked out to the User.

The Link Info Result Set is defined using T-SQL syntax as described in the Common Result Sets Single Doc Link Information Result Set (section 2.2.5.19).

3.1.5.43.7 Contained Document Metadata Result Set

The Contained Document Metadata Result Set contains the metadata information for the Documents contained within the specified document. If the specified document is not a folder, the Contained Document Metadata Result Set MUST NOT be returned. Otherwise, the Contained Document Metadata Result Set MUST return one row for each contained document in the site collection which has its directory name equal to the value of the **@DirFullUrl** parameter, and is a published or draft version which is not checked out to the specified User, or is a checked out version which is checked out to the User, and either the contained document has a Document Store Type (section 2.2.2.4) of 0 (File) and the **@IncludeListItems** parameter is set to 1, the document is in a document library, or the document can have a document stream, or the contained document is a folder, and the folder is not a thicket supporting file, and either the folder is not a thicket folder or the **@IncludeThicketDirs** parameter is set to 1.

```
Id                          uniqueidentifier,
{FullUrl}                   nvarchar(260),
Type                        tinyint,
MetaInfoTimeLastModified    datetime,
MetaInfo                    image,
Size                        int,
TimeCreated                 datetime,
```


Copyright © 2014 Microsoft Corporation.

```
TimeLastModified              datetime,
Version                       int,
DocFlags                      int,
{ListType}                    int,
tp_Name                       nvarchar(38),
{ListTitle}                   nvarchar(255),
CacheParseId                  uniqueidentifier,
{GhostDirName}                nvarchar(256),
{GhostLeafName}               nvarchar(128),
tp_Login                      nvarchar(255),
CheckoutDate                  datetime,
{CheckoutExpires}             datetime,
VirusStatus                   int,
VirusInfo                     nvarchar(255),
SetupPathVersion              tinyint,
SetupPath                     nvarchar(255),
SetupPathUser                 nvarchar(255),
NextToLastTimeModified        datetime,
UIVersion                     int,
CheckinComment                nvarchar(1023),
WelcomePageUrl                nvarchar(260),
WelcomePageParameters         ntext,
tp_Flags                      bigint,
Acl                           image,
AnonymousPermMask             bigint,
DraftOwnerId                  int,
Level                         tinyint,
ParentVersion                 int,
TransformerId                 uniqueidentifier,
ParentLeafName                nvarchar(128),
ProgId                        nvarchar(255),
DoclibRowId                   int,
tp_DefaultWorkflowId          uniqueidentifier,
ListId                        uniqueidentifier;
```

Id: The Document Identifier (section 2.2.1.2) of this contained document.

{FullUrl}: The store-relative form URL for this document.

Type: The Document Store Type of this document.

MetaInfoTimeLastModified: A datetime with a timestamp in UTC format specifying the last time the Metainfo value of this document was changed. This value can be NULL.

MetaInfo: A metadict for this document. The metadict format is specified in [MS-FPSE] section 2.2.2.2.11. This value can be NULL, and MUST be NULL if the MetaInfoTimeLastModified value is not more recent than the **@ClientTimeStamp** parameter.

Size: The number of bytes in the document stream of this document. This value can be NULL.

TimeCreated: A datetime with a timestamp in UTC format specifying when this document was created.

TimeLastModified: A datetime with a timestamp in UTC format specifying when this document was last modified.

Version: A counter incremented any time a change is made to this document, and used for internal conflict detection.

Copyright © 2014 Microsoft Corporation.

DocFlags: A Doc Flags (section 2.2.2.3) value describing this document. This value can be NULL.

{ListType}: A packed combination of the List Base Type (section 2.2.3.11) and List Server Template (section 2.2.3.12) values of the List containing this document, consisting of the List Server Template value multiplied by 256 and added to the value of the List Base Type.

tp_Name: The Identifier of the List that contains this document.

{ListTitle}: If the containing document is the root folder of the List which contains this document, this contains the display name of the List. Otherwise, this value MUST be NULL.

CacheParseId: This value MUST be NULL.

{GhostDirName}: This value MUST be NULL.

{GhostLeafName}: This value MUST be NULL.

tp_Login: If this document is currently checked out, this is the login name of the User to whom it is checked out. In all other cases, this MUST be NULL.

CheckoutDate: A datetime with a timestamp in UTC format specifying when this document was checked out. This value can be NULL.

{CheckoutExpires}: A datetime with a timestamp in UTC format specifying when the short-term lock for this document will expire. If this date is in the past, this document SHOULD be treated as unlocked. This value can be NULL if no short-term lock has been placed on the document. This value MUST be NULL if a User has this document checked out.

VirusStatus: A Virus Status (section 2.2.3.18) value specifying the current virus state of this document. This value can be NULL if this document has not been processed by a virus scanner.

VirusInfo: A string containing a provider-specific message returned by the virus scanner when it last processed this document. This value can be NULL if this document has not been processed by a virus scanner.

SetupPathVersion: If this is a ghosted document, this specifies the setup path location that the **SetupPath** fragment is relative to. This value MUST NOT be NULL. The following are valid values.

Value	Description
2	The SetupPath is relative to the install location of Windows SharePoint Services 2.0 on the WFE (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\60)
3	The SetupPath is relative to the install location of Windows SharePoint Services 3.0 on the WFE (for example, Program Files\Common Files\Microsoft Shared\Web Server Extensions\12).

SetupPath: If this document is now or once was ghosted, this contains the setup path fragment relative to the base setup path described above by the **SetupPathVersion** value, where the content stream of this document can be found. This value can be NULL.

SetupPathUser: If this document is now or once was ghosted, this contains the login name of the User that created the ghosted document. This value can be NULL.

NextToLastTimeModified: The value of **TimeLastModified** from the previous time when the document was last saved. If the **NextToLastTimeModified** value matches the **TimeLastModified** value when the change occurred and the client has a version of this document that it has

Copyright © 2014 Microsoft Corporation.

successfully modified, the client can safely submit the document to the server despite what appears to be an intervening edit to this document. This value can be NULL.

UIVersion: The UI version number for this document.

CheckinComment: An optional user-supplied description provided when this document is checked in or published. This value can be NULL.

WelcomePageUrl: If this document is a folder, this specifies an optional page to redirect to when the folder is requested with an HTTP GET operation. The URL is relative to the URL of the folder itself, and MUST be contained within that folder. Attempts to break out of the folder, such as "../../somepage.aspx", are not valid. This value can be NULL.

WelcomePageParameters: Contains optional "search" parameters to specify as part of the **WelcomePageUrl** value. These parameters start at either the query string signifier '?' or the bookmark signifier '#'. This value can be NULL.

tp_Flags: The List Flags (section 2.2.2.5) value for the List that contains this document.

Acl: The binary serialization of the WSS ACL for this document. The WSS ACL is either explicitly defined for this document, or inherited from the parent object of this document. This value can be NULL.

AnonymousPermMask: A WSS Rights Mask that indicates the rights granted to a User that is anonymous, or has no specific rights, to this document. This value can be NULL.

DraftOwnerId: The User Identifier (section 2.2.1.13) of the User that published this document as a draft. This value MUST be NULL if the document is not a draft version.

Level: A Publishing Level Type (section 2.2.2.6) value specifying the publishing status of this document.

ParentVersion: If this document is a transformed version of another document, this is the UI version value from the original document. This value MUST be NULL if the document is not the product of a document transformation.

TransformerId: If this document is a transformed version of another document, this is the GUID of the agent that performed the transformation. This value MUST be NULL if the document is not the product of a document transformation.

ParentLeafName: If this document is a transformed version of another document, this is the leaf name of the original document. The original document MUST be in the same folder as the transformed document. If either is moved, the relationship is broken. This value MUST be NULL if the document is not the product of a document transformation.

ProgId: Specifies a preferred application to open this document. The ProgId value is used to distinguish between different applications that save files with a given file extension (for example, different editors for .HTML or .XML files). This value MUST be NULL if a ProgId was not specified when this document was saved.

DoclibRowId: The document library row identifier for this document. If this document is not contained in a List, this value MUST be NULL.

tp_DefaultWorkflowId: The Workflow Identifier (section 2.2.1.16) corresponding to the workflow (2) to be invoked if this document is in a moderated List and this document is submitted for approval as part of a check in.

Copyright © 2014 Microsoft Corporation.

ListId: The List Identifier (section 2.2.1.5) of the List that contains this document. If this document is not contained in a List, this value MUST be NULL.

3.1.5.44 proc_RenameUrl

The **proc_RenameUr**l stored procedure is invoked to move a document (update its URL) within a specified site (2).

```
PROCEDURE proc_RenameUrl(
      @SiteId                    uniqueidentifier,
      @SubWebId                  uniqueidentifier,
      @OldUrl                    nvarchar(260),
      @NewUrl                    nvarchar(260),
      @UserId                    int,
      @RenameFlags               int               = 0,
      @PutFlags                  int               = 0,
      @ReturnFlags               int               = 0,
      @AttachmentOpOldUrl        int               = 3,
      @AttachmentOpNewUrl        int               = 3,
      @ParseDocsNow              tinyint           = NULL     OUTPUT,
      @FailedUrl                 nvarchar(260)     = NULL     OUTPUT
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the document.

@SubWebId: The Site Identifier (section 2.2.1.11) of the Site containing the document.

@OldUrl: The URL of the document to be moved.

@NewUrl: The target URL of the document after the move.

@UserId: The User Identifier (section 2.2.1.13) of the user who is requesting the move.

@RenameFlags: A bit field (2) determining document rename options. This can have one or more flags set. The only valid values of the **@RenameFlags** bits are specified in Rename Flags (section 2.2.2.8).

@PutFlags: A bit field (2) determining document change options. This can have one or more flags set. The only valid values of the **@PutFlags** bits are specified in Put Flags Type (section 2.2.2.7).

@ReturnFlags: A bit field (2) determining the type of information requested. This can have one or more flags set. The only valid bit values of **@ReturnFlags** are specified as follows.

Value	Description
0x01	Return information about the moved Documents.
0x02	Return information about backward links referencing the moved Documents.

@AttachmentOpOldUrl: An Attachments Flag (section 2.2.3.1) value for the document URL.

@ AttachmentOpNewUrl: An Attachments Flag value for the destination URL.

@ParseDocsNow: An output parameter set to 1 to indicate that the moved document MUST be parsed again. Reparsing is necessary when the document properties need to be scanned, for

Copyright © 2014 Microsoft Corporation.

example, when the document is moved across lists, or when the Document's file extension is modified.

@FailedUrl: An output parameter indicating the URL at which the move operation failed. Set to NULL if the move was successful.

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

Value	Description
0	Successful execution.
2	File not found: Parent of target URL not found.
3	Path not found: Item not found at specified URL, site collection and Site.
5	Access denied: The User specified in @UserId lacks the necessary privileges.
15 or 51	Cannot move to, or from, a forms folder in a List or document library.
33	Cannot move a folder which contains checked out Documents.
34	Cannot rename a folder within a List or move a folder out of a List when the List is not a document library
50	The document is a Site, but @RenameFlags does not indicate that a Site move is requested.
80	Target URL is a document with a Document Store Type (section 2.2.2.4) of 0 (File), but @PutFlags value does not indicate a request to overwrite it.
87	Bad parameter: Unspecified Error prevented the requested move.
130	Cannot move to, or from, an image or a thumbnail file in an Image Library.
138	Cannot move Folder across Lists.
144	Not an overwrite request and URL is a directory or web, or other invalid combination.
161	Cannot move Folder across Sites.
190	Cannot move Folder containing thicket.
206	Target URL is too long.
214	Cannot move thicket.
1150	Concurrency Violation.
1358	Internal database corruption.
1359	Unknown Internal Error.
8398	There was an error moving a List. At least one List could not be deleted.

The **proc_RenameUrl** stored procedure MUST return either zero, one or two result sets, depending on the values for input parameters as described for each result set.

Copyright © 2014 Microsoft Corporation.

3.1.5.44.1 Rename Result Set

The Rename Result Set returns basic information about the old and new URLs for all moved (renamed) Documents. When renaming a container object, affected items in the container are included in the Rename Result Set. The Rename Result Set MUST be returned only when requested (when bit 0x01 is set in **@ReturnFlags**). If the Rename Result Set is returned, it MUST return one row for each document which was renamed during the operation.

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

{OldUrlDirName}: The directory name of the document before rename.

{OldUrlLeafName}: The leaf name of the document before rename.

{NewUrlDirName}: The directory name of the document after rename.

{NewUrlLeafName}: The leaf name of the document after rename.

{Type}: The Document Store Type (section 2.2.2.4) of the renamed document.

3.1.5.44.2 Backward Link Result Set

The Backward Link Result Set returns the URL of each document containing a backward link to the moved (renamed) document(s). The Backward Link Result Set MUST be returned only when requested (bit 0x2 is set in **@ReturnFlags** and bit 0x4 is set in **@RenameFlags**.) If the Backward Link Result Set is returned, it MUST return one row for each item containing a backward link to any of the renamed document(s).

```
DocUrl                       nvarchar(260),
```

DocUrl: The store-relative form URL to the document holding the backward link to the renamed document.

3.1.5.45 proc_SecAddPrincipalToRole

The **proc_SecAddPrincipalToRole** stored procedure is invoked to add a **security principal** (2) to a role defined within a site collection.

```
PROCEDURE proc_SecAddPrincipalToRole(
      @SiteId                    uniqueidentifier,
      @WebId                     uniqueidentifier,
      @ScopeId                   uniqueidentifier,
      @RoleId                    int,
      @UserId                    int,
      @AddChangeLog              bit = 0,
      @ReturnAuditMask           bit = 1
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the role and principal (1).

Copyright © 2014 Microsoft Corporation.

@WebId: The Site Identifier (section 2.2.1.11) of the site (2) containing the role and principal (1).

@ScopeId: The Scope Identifier (section 2.2.1.8) of the scope containing the role.

@RoleId: Specifies the Role Identifier (section 2.2.1.7) of the **role definition** that the principal (1) is added into. **@RoleId** MUST correspond to a valid role.

@UserId: The User Identifier (section 2.2.1.13) for the principal (1) to be added to the specified role. This value MUST refer to an existing User Identifier for the specified site collection.

@AddChangeLog: A flag to specify whether **proc_SecAddPrincipalToRole** updates the change log. A value of 1 indicates that the procedure MUST update the change log. The default value of 0 indicates that the procedure MUST NOT update the change log.

@ReturnAuditMask: A flag specifying whether to return Audit Flags (section 2.2.2.1) data for the specified Site. The default value of 1 indicates that **proc_SecAddPrincipalToRole** MUST return the Audit Flags data. A value of 0 indicates that **proc_SecAddPrincipalToRole** MUST NOT return the Audit Flags data.

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

Value	Description
0	Successful execution.
3	The Site specified by **@WebId** is not valid for membership permissions or there are no roles defined for this Site.

The **proc_SecAddPrincipalToRole** stored procedure MUST return one result set if the **@ReturnAuditMask** value is set to 1, and MUST NOT return a result set if the value is 0.

3.1.5.45.1 Site Audit Mask Result Set

The Site Audit Mask Result Set contains information about the Audit Flags (section 2.2.2.1) set for the site (2). The Site Audit Mask Result Set MUST be returned if audit information was requested by setting the @ReturnAuditMask value to 1. The Site Audit Mask Result Set is defined in the Common Result Sets Site Audit Mask Result Set (section 2.2.5.20).

3.1.5.46 proc_SecAddRoleDef

The **proc_SecAddRoleDef** stored procedure creates a new role definition for a specified site (2) within a site collection.

```
PROCEDURE proc_SecAddRoleDef (
      @SiteId                       uniqueidentifier,
      @WebId                        uniqueidentifier,
      @Title                        nvarchar(255),
      @Description                  nvarchar(512),
      @Hidden                       bit,
      @RoleOrder                    int,
      @Type                         tinyint,
      @PermMask                     bigint,
      @IdToCreate                   int,
      @RoleDefId                    int = NULL OUTPUT
);
```


Copyright © 2014 Microsoft Corporation.

@SiteId: The Site Collection Identifier (section 2.2.1.9) for the site collection to add the role definition to.

@WebId: The Site Identifier (section 2.2.1.11) for the Site to add the role definition to. The Site MUST have its own scope specifying unique permissions.

@Title: The title of the role definition for display in the WFE. This MUST be a value which does not match an existing role definition title within the Site, and MUST NOT be NULL or an empty string.

@Description: The description of the role definition for display in the WFE.

@Hidden: A bit flag specifying whether or not the role definition is to be displayed in the WFE. This value MUST NOT be NULL. If this parameter is set to 1, the WFE MUST suppress display of the role definition in the role management WFE.

@RoleOrder: An integer value specifying the relative position in which this role definition is to be displayed in the WFE. Multiple role definitions can have the same **@RoleOrder** value. This value MUST NOT be NULL. The WFE uses this value to determine the order in which role definitions are displayed on the Permission Levels page for the site (2). Role definition items displayed are sorted in ascending order by **@RoleOrder**, then by descending role type specified by **@Type**.

@Type: The Role Definition Type (section 2.2.3.16) value for the role definition to be added.

@PermMask: A WSS Rights Mask (section 2.2.2.13) specifying the rights to grant to the role definition. The WSS Rights Mask associated with the role definition MUST be set to this value.

@IdToCreate: The Role Identifier (section 2.2.1.7) to assign to the new role definition. If this parameter is NULL, a new value MUST be assigned and returned in **@RoleDefId**. This parameter MUST NOT be a Role Identifier identifier which already exists within the site collection.

@RoleDefId: The Role Identifier of the created role definition, returned as an output parameter.

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

Value	Description
0	Successful execution.
80	The @IdToCreate or @Title already exists or is NULL for a role definition within the site collection.
1816	The limit for the maximum number of role definitions in the specified site (2) has been reached.

Upon successful completion, the **proc_SecAddRoleDef** stored procedure MUST return a single result set as specified in the following.

3.1.5.46.1 Site Audit Mask Result Set

The Site Audit Mask Result Set contains information about the Audit Flags (section 2.2.2.1) associated with the Site. The Site Audit Mask Result Set MUST be returned on successful completion and MUST contain one row. The Site Audit Mask Result Set is defined in the Common Result Sets Site Audit Mask Result Set (section 2.2.5.20).

3.1.5.47 proc_SecAddUser

The **proc_SecAddUser** stored procedure is invoked to add an entry for a principal (1) (a user or domain group) to the list of user information stored in the BEDS.

Copyright © 2014 Microsoft Corporation.

```
PROCEDURE proc_SecAddUser (
      @SiteId                     uniqueidentifier,
      @SystemId                   varbinary(512),
      @ExternalToken              image,
      @ExternalTokenTime          datetime,
      @IsDomainGroup              bit,
      @IsActive                   bit,
      @Login                      nvarchar(255),
      @Title                      nvarchar(255),
      @Email                      nvarchar(255),
      @Notes                      nvarchar(1023),
      @MembershipWebId            uniqueidentifier = NULL,
      @IncrementUserCount         bit = 0,
      @ImportDeleted              bit = 0,
      @AddedToTable               bit OUTPUT,
      @UserIdOut                  int OUTPUT,
      @LoginOut                   nvarchar(255) OUTPUT,
      @TitleOut                   nvarchar(255) OUTPUT,
      @EmailOut                   nvarchar(255) OUTPUT,
      @NotesOut                   nvarchar(1023) OUTPUT,
      @DeletedOut                 bit OUTPUT
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection to associate with the principal (1).

@SystemId: The **SystemID** (section 2.2.1.12) of the principal (1) to be added or updated. If a user exists with the specified **SystemID**, its record will be updated. Otherwise, a new User will be added with the specified **SystemID**.

@ExternalToken: An External Group Token (section 2.2.4.2) encoding information on the principal's domain group membership, derived from an external role provider.

@ExternalTokenTime: A datetime in UTC format specifying the most recent time the @ExternalToken value was updated.

@IsDomainGroup: A bit flag specifying whether the principal (1) to be added is a domain group. If this is set to "1", the principal (1) being added is a domain group. A value of "0" indicates this is a User. This parameter MUST NOT be NULL.

@IsActive: A bit specifying if the principal (1) is an active user in the site collection. When set to "1", the principal (1) is an active user in the site collection. Otherwise, the principal (1) is NOT an active user in the site collection. This flag MUST NOT be NULL.

@Login: The login name of the principal (1) to be added. This parameter MUST NOT be NULL.

@Title: The display name of the principal (1) to be added. This parameter MUST NOT be NULL.

@Email: The email address of the principal (1) to be added. This parameter MUST NOT be NULL.

@Notes: A text string containing notes about the principal (1) to be added. This parameter MUST NOT be NULL.

@MembershipWebId: The Site Identifier (section 2.2.1.11) of the site (2) to associate with the principal to be added.

Copyright © 2014 Microsoft Corporation.

@IncrementUserCount: A bit flag specifying whether to increment the User count of the site collection. When this parameter is set to "1", **proc_SecAddUse**r MUST increment the site collection User count.

@ImportDeleted: A bit flag specifying whether the principal (1) is to be added as deleted. When this parameter is set to "1**", proc_SecAddUser** MUST mark the User information for the principal (1) to be added as deleted. The deleted state is set within the User information, rather than dropping entries from the User information table, to preserve list item ownership information.

@AddedToTable: This is an output parameter indicating that an entry has been added to the User information table for the principal (1). Its value MUST be listed in the following table.

Value	Description
0	The principal (1) already exists in the User information table.
1	The User information for the principal (1) has been added to the table.

@UserIdOut: An output parameter which MUST contain the User Identifier (section 2.2.1.13) of the added principal (1) on successful completion.

@LoginOut: An output parameter which MUST contain the login name of the added principal (1) on successful completion.

@TitleOut: An output parameter which SHOULD contain the display name of the added principal (1) on successful completion. MAY contain an empty string if no display name is available.

@EmailOut: An output parameter which SHOULD contain the email address of the added principal (1) on successful completion. MAY contain an empty string if no email address is available.

@NotesOut: An output parameter which SHOULD contain the notes about the added principal (1) on successful completion. MAY contain an empty string if no notes content is available.

@DeletedOut: An output parameter which can indicate the deleted state of the principal (1). Its value MUST be listed in the following table.

Value	Description
0	The User information has not been marked as deleted.
1	The User information has been marked as deleted.

Return Values: The **proc_SecAddUser** stored procedure returns an integer return code. Its value MUST be listed in the following table.

Value	Description
0	Successful execution.
80	A principal (1) with the same login name but a different **SystemID** already has an entry in the User information.
1816	User Quota is exceeded and no additional users can be added to the site (2).

The **proc_SecAddUser** stored procedure MUST NOT return any result sets.

Copyright © 2014 Microsoft Corporation.

3.1.5.48 proc_SecAddUserToSiteGroup

The **proc_SecAddUserToSiteGroup** stored procedure is invoked to add a user to a site group in the site collection. The User is added to the site group and the root site of the site collection, the user's User token and status as active user is updated, and the change is logged to the change log and site audit log.

```
PROCEDURE proc_SecAddUserToSiteGroup(
      @SiteId                       uniqueidentifier,
      @GroupId                      int,
      @UserIDToBeAdded              int,
      @UserID                       int,
      @SiteAdmin                    bit,
      @BelongsToGroup               bit,
      @GroupOwnerId                 int,
      @CurrentUserIsOwner           bit
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) for a site collection that will be queried for the requested Group.

@GroupId: An identifier for the site group that the User (specified by **@UserIDToBeAdded**) refers to. **@GroupId** MUST refer to an existing group.

@UserIDToBeAdded: A User Identifier (section 2.2.1.13) for the User being added to the specified group. This value MUST refer to an existing User Identifier for the specified site collection.

@UserID: The User Identifier for the current user who is adding the User to the specified group. This value MUST refer to an existing User Identifier for the specified site collection.

@SiteAdmin: If 1, the User is a site administrator. Otherwise, the User is not.

@BelongsToGroup: If 1, the User belongs to the site group. Otherwise, the User does not.

@GroupOwnerId: An identifier for the User who is the owner of the site group specified by **@GroupId**. **@GroupOwnerId** MUST refer to an existing owner, which can be either a user or another group.

@CurrentUserIsOwner: If 1, the User specified by **@GroupOwnerId** is the owner of the site group specified by **@GroupId**. Otherwise, the User is not the owner.

Return Values: The **proc_SecAddUserToSiteGroup** stored procedure returns an integer return code. The following are valid values.

Value	Description
0	Successful execution.
5	The User specified by @UserID does not have rights to modify Group membership of the Group specified by @GroupID.

The **proc_SecAddUserToSiteGroup stored** procedure MUST NOT return any result sets.

Copyright © 2014 Microsoft Corporation.

3.1.5.49 proc_SecAddWebMembership

The **proc_SecAddWebMembership** stored procedure is invoked to add a user to the membership of a site (2).

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

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the requested site (2). This parameter MUST be ignored.

@WebId: The Site Identifier (section 2.2.1.11) of the site (2) to add a new User to.

@UserId: The User Identifier (section 2.2.1.13) for the User to be added to site membership. This value MUST refer to an existing User Identifier for the specified site collection

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

Value	Description
0	Successful execution.
3	The Site specified by @WebId was not found, or the Site defining its permissions was not found.

The **proc_SecAddWebMembership** stored procedure MUST NOT return a result set.

3.1.5.50 proc_SecChangeToInheritedList

The **proc_SecChangeToInheritedList** stored procedure is invoked to change the scope of the specified list (1) to the specified site (2) and remove any role assignments and permission settings specific to the List.

The permissions of any List Items contained in the specified List which have unique permissions MUST NOT be changed.

```
PROCEDURE proc_SecChangeToInheritedList(
      @WebId                        uniqueidentifier,
      @ListId                       uniqueidentifier
);
```

@WebId: The Site Identifier (section 2.2.1.11) of the Site containing the List.

@ListId: The List Identifier (section 2.2.1.5) of the List.

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

Value	Description
0	Successful execution.

Copyright © 2014 Microsoft Corporation.

Value	Description
3	The List was not found, or the List does not have unique permissions.

The **proc_SecChangeToInheritedList** stored procedure MUST return one result set on successful completion. Otherwise, no result sets are returned.

3.1.5.50.1 Site Audit Mask Result Set

The Site Audit Mask Result Set contains information about the Audit Flags (section 2.2.2.1) associated with the specified Site. The Site Audit Mask Result Set MUST be returned on successful completion, and MUST return a single row. The Site Audit Mask Result Set is defined in the Common Result Sets Site Audit Mask Result Set (section 2.2.5.20).

3.1.5.51 proc_SecChangeToInheritedWeb

The **proc_SecChangeToInheritedWeb** stored procedure changes a site (2) from having its own unique permissions to instead use the permissions inherited from its nearest ancestor with unique permissions. When a Site is changed to have inherited permissions, all role definitions and fine-grained permissions on its lists and document libraries are reset.

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

@SiteId: The Site Collection Identifier (section 2.2.1.9) for the site collection containing the site (2) to be changed.

@WebId: The Site Identifier (section 2.2.1.11) for the Site to change to use inherited permissions.

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

Value	Description
0	Successful execution.
3	The Site was not found, the Site does not have unique permissions, or an ancestor Site with unique permissions was not found.

The **proc_SecChangeToInheritedWeb** stored procedure MUST return two result sets on successful completion. Otherwise no result sets are returned.

3.1.5.51.1 Inherited Site Result Set

The Inherited Site Result Set returns the scope and Site Identifiers (section 2.2.1.11) that the site (2) specified by @WebId now inherits its permissions from. The Inherited Site Result Set MUST be returned on successful completion and MUST contain one row.

```
      ScopeId                       uniqueidentifier,
      {ParentWebPermAncestor}       uniqueidentifier;
```


Copyright © 2014 Microsoft Corporation.

ScopeId: The Scope Identifier (section 2.2.1.8) for the scope that the changed site (2) specified by **@WebId** now inherits from.

{ParentWebPermAncestor}: The Site Identifier for the site (2) that the changed site specified by **@WebId** now derives permissions from.

3.1.5.51.2 Site Audit Mask Result Set

The Site Audit Mask Result Set contains information about the Audit Flags (section 2.2.2.1) associated with the specified Site. The Site Audit Mask Result Set MUST be returned on successful completion and MUST contain a single row.

The Site Audit Mask Result Set is defined in the Common Result Sets Site Audit Mask Result Set (section 2.2.5.20).

3.1.5.52 proc_SecChangeToUniqueScope

The **proc_SecChangeToUniqueScope** stored procedure sets a securable object such as a site (2), list (1), or document library to use its own unique scope, instead of inheriting its scope.

```
PROCEDURE proc_SecChangeToUniqueScope(
      @SiteId                       uniqueidentifier,
      @WebId                        uniqueidentifier,
      @OldScopeId                   uniqueidentifier,
      @CopyFromScopeId              uniqueidentifier,
      @Url                          nvarchar(260),
      @DocId                        uniqueidentifier,
      @bIsWeb                       bit,
      @UserId                       int,
      @CopyAnonymousMask            bit,
      @CopyRoleAssignments          bit,
      @bBreakBySiteOwner            bit,
      @ReturnAuditMask              bit,
      @NewScopeId                   uniqueidentifier = NULL OUTPUT
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the securable object to be set to use a unique scope.

@WebId: The Site Identifier (section 2.2.1.11) of the site (2) that is or contains the securable object.

@OldScopeId: The Scope Identifier (section 2.2.1.8) for the original scope of the securable object specified by **@Url** or **@DocId**.

@CopyFromScopeId: The Scope Identifier for a scope to copy the administrator role, anonymous user permissions, and role assignments from for use as the new scope. This parameter MUST NOT be NULL.

@Url: The store-relative form URL for the securable object. The **@Url** parameter MUST be NULL to specify the securable object with the **@DocId** parameter. The securable object MUST be specified by the **@Url** or the **@DocId** parameter. The @Url parameter MUST be NULL to specify the securable object with the **@DocId** parameter.

@DocId: The Document Identifier (section 2.2.1.2) of the securable object. The **@DocId** parameter MUST be ignored and MAY be NULL if **@Url** specifies the securable object.

Copyright © 2014 Microsoft Corporation.

@bIsWeb: A bit flag specifying whether the securable object is a site (2). If this parameter is set to 1, the securable object is a site; all subsites which inherit their scope MUST have their inheritances reset to the new scope. This parameter MUST NOT be set to 1 if **@Url** does not point to a site (2), and this parameter MUST be set to 1 if **@Url** does point to a site.

@UserId: Specifies the principal (1) to be added to the new scope in the administrator role, unless overridden by the **@bBreakBySiteOwner** parameter or the **@CopyFromScopeId** parameter.

@CopyAnonymousMask: A bit flag specifying whether to copy anonymous user permissions from the **@CopyFromScopeId** parameter into the new scope. If this parameter is set to 1, proc_SecChangeToUniqueScope MUST copy the permissions for anonymous user access from the **@CopyFromScopeId** parameter into the new scope.

@CopyRoleAssignments: A bit flag specifying whether to copy the role assignments from the **@CopyFromScopeId** parameter into the new scope. If this parameter is set to 1, proc_SecChangeToUniqueScope MUST copy the role assignments from the **@CopyFromScopeId** parameter into the new scope. If both **@bBreakBySiteOwner** and **@CopyRoleAssignments** are set, then the setting of CopyRoleAssignments will take precedence. Role assignments are copied from the scope provided in **@CopyFromScopeId**.

@bBreakBySiteOwner: A bit flag specifying whether to use the site owner in the administrator role instead of the principal (1) specified by **@UserId**. If this parameter is set to 1, proc_SecChangeToUniqueScope MUST set the Site owner in the administrator role in the new scope definition. If both **@**bBreakBySiteOwner and **@CopyRoleAssignments** are set, then the setting of CopyRoleAssignments will take precedence. Role assignments are copied from the scope provided in **@CopyFromScopeId**.

@ReturnAuditMask: A bit flag specifying whether to return a Site Audit Mask Result Set. If this parameter is set to 1, proc_SecChangeToUniqueScope MUST return a Site Audit Mask Result Set on successful completion.

@NewScopeId: The Scope Identifier generated for the new scope, returned as an output parameter.

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

Value	Description
0	Successful execution.
3	The specified securable Object was not found at the specified location, the **@OldScopeId** does not match the securable object's ScopeId, or the securable object has unique permissions.

If **@ReturnAuditMask** has a value of 1, the **proc_SecChangeToUniqueScope** stored procedure MUST return a single result set on successful completion; otherwise, zero result sets MUST be returned.

3.1.5.52.1 Site Audit Mask Result Set

The Site Audit Mask Result Set contains information about the Audit Flags associated with the Site. If @ReturnAuditMask has a value of 1, the Site Audit Mask Result Set MUST be returned on successful completion, and MUST contain only one row. The Site Audit Mask Result Set is defined in the Common Result Sets Site Audit Mask Result Set (section 2.2.5.20).

Copyright © 2014 Microsoft Corporation.

3.1.5.53 proc_SecCheckDeletedAccounts

The **proc_SecCheckDeletedAccounts** stored procedure is invoked to check if a login name exists in the site collection.

```
PROCEDURE proc_SecCheckDeletedAccounts(
      @SiteId                       uniqueidentifier,
      @Login                        nvarchar(255)
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) for the site collection.

@Login: The login name the principal (1) as specified by the authentication in use. This parameter MUST NOT be NULL.

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

The **proc_SecCheckDeletedAccounts** stored procedure MUST return a single result set as follows:

3.1.5.53.1 Login Result Set

If a valid login name is found in the site collection, the Login Result Set MUST return one or more rows containing the Login information (one row for each time **@**Login is found UserInfo Table (section 2.2.7.10)); otherwise zero rows are returned.

```
      tp_Login                      nvarchar(255);
```

tp_Login: The login name for the user.

3.1.5.54 proc_SecCloneRoleDefinitions

The **proc_SecCloneRoleDefinitions** stored procedure creates a copy of the current role definition for a site (2). After successful execution, the Site will have its own copy of the role definition and unique role assignments, and the Site and all its subsites will use the new role definition on this Site (unless the subsite has its own role definition). The new Scope Identifier (section 2.2.1.8) is returned in an output parameter. proc_SecCloneRoleDefinitions also returns a result set containing the Site Audit Flags (section 2.2.2.1) and masks.

```
PROCEDURE proc_SecCloneRoleDefinitions(
      @SiteId                       uniqueidentifier,
      @WebId                        uniqueidentifier,
      @CopyRoleAssignments          bit,
      @UserId                       int,
      @NewScopeId                   uniqueidentifier   OUTPUT
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the Site.

@WebId: The Site Identifier (section 2.2.1.11) of the site (2) which will have a copy of its current role definition.

@CopyRoleAssignments: Specifies whether to keep the current role assignments. **@CopyRoleAssignments** MUST NOT be NULL:

Copyright © 2014 Microsoft Corporation.

- If set to 1, **proc_SecCloneRoleDefinitions** will keep the current role assignments.
- If set to 0, **proc_SecCloneRoleDefinitions** will add the user specified by **@UserId** to the Administrator Role, and remove everyone else from all roles.

@UserId: The User Identifier (section 2.2.1.13) of the current user. **@UserId** is assigned to the Administrator Role when **@**CopyRoleAssignments is 0. It is also used for security event logging. This value MUST refer to an existing User Identifier for the specified site collection.

@NewScopeId: An output parameter which contains a Scope Identifier for the scope of the site (2). If the site (2) already has unique role assignments before this call, a new scope MUST NOT BE generated, and the output parameter MUST be the site's original Scope Identifier. If the Site does not have unique role assignments before this call, a new Security Scope MUST be generated for this Site, and the ID of the new Security Scope MUST be returned in the output parameter.

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

Value	Description
0	Successful execution.
3	Site not found. Either the Site specified by @SiteId/@WebId does not exist, or the Site already has its own role definition.

Upon successful execution, the **proc_SecCloneRoleDefinitions** stored procedure MUST return a single result set as follows.

3.1.5.54.1 Site Audit Mask Result Set

The Site Audit Mask Result Set contains the information about the Audit Flags (section 2.2.2.1) associated with the specified Site. On successful execution, the Site Audit Mask Result Set MUST return once if **@**CopyRoleAssignments is set to "1". If **@**CopyRoleAssignments is set to "0", the Site Audit Mask Result Set MUST return twice on successful execution.

The Site Audit Mask Result Set MUST contain a single row.

3.1.5.55 proc_SecCreateSiteGroup

The **proc_SecCreateSiteGroup** stored procedure is invoked to add a new site group to a site collection.

```
PROCEDURE proc_SecCreateSiteGroup(
      @SiteId                       uniqueidentifier,
      @Title                        nvarchar(255),
      @Description                  nvarchar(512),
      @OwnerID                      int,
      @OwnerIsUser                  bit,
      @FirstMemberId                int,
      @UseExisting                  bit,
      @SelfOwner                    bit,
      @GroupID                      int               OUTPUT
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.11) of the site collection containing the new site group to be created.

Copyright © 2014 Microsoft Corporation.

@Title: The title of the new site group. This parameter MUST NOT be NULL.

@Description: A description of the new site group.

@OwnerID: A User Identifier (section 2.2.1.13) or Site Group Identifier (section 2.2.1.10) for the new site group's owner.

@OwnerIsUser: A bit flag specifying whether the site group owner specified by **@OwnerID** is a user or a site group.

- When **@OwnerIsUser** is set to 1, the new site group owner is a User in the site collection.
- When **@OwnerIsUser** is set to 0, the owner is a site group

@FirstMemberId: A User Identifier for the first member of the new site group. This value MUST correspond to an existing user id or be NULL. If this value is NULL then the site group MUST be created with no members.

@UseExisting: A bit flag specifying whether to return the Site Group Identifier of an existing site group with a title matching **@Title**.

- When **@UseExisting** is set to 1, the procedure MUST attempt to locate an existing site group with the same title as the **@Title** input parameter, and if such a site group is found, return without creating a new site group or modifying the existing site group.
- When **@UseExisting** is set to 0, the procedure MUST return a failure return code value of 80 if an existing site group with a matching title is found.

@SelfOwner: A bit flag specifying whether the site group is its own owner. When **@SelfOwner** is set to 1, the site group MUST be set as its own owner, and the values of **@OwnerID** and **@OwnerIsUser** MUST be ignored.

@GroupID: An output parameter holding the integer Site Group Identifier for the newly created or existing site group.

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

Value	Description
0	Successful execution.
80	Failed to create the site group because a site group with an identical title exists in the site collection.

The **proc_SecCreateSiteGroup** stored procedure MUST return no result sets.

3.1.5.56 proc_SecDecCurrentUsersCount

The **proc_SecDecCurrentUsersCount** stored procedure is invoked to reduce by one the total number of users in the specified site collection when configured to use Active Directory Creation Mode. **proc_SecDecCurrentUsersCount** does not delete any User information.

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


Copyright © 2014 Microsoft Corporation.

@SiteID: The Site Collection Identifier (section 2.2.1.9) of the site collection.

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

The **proc_SecDecCurrentUsersCount** stored procedure MUST NOT return any result sets.

3.1.5.57 proc_SecGetAccountStatus

The **proc_SecGetAccountStatus** stored procedure provides status information for a site collection's users matching a specified login name or email address, which are not marked as deleted.

```
PROCEDURE proc_SecGetAccountStatus(
      @SiteId                       uniqueidentifier,
      @Login                        nvarchar(255),
      @Email                        nvarchar(255)
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the specified Users.

@Login: The login name of a User to be matched. If the **@Email** parameter specifies a non-empty email address, the **@Login** parameter can be NULL or an empty string. Otherwise, this parameter MUST NOT be NULL.

@Email: The email address of a User to be matched. If the **@Login** parameter specifies a non-empty login name, the **@Email** parameter can be NULL or an empty string.

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

The **proc_SecGetAccountStatus** stored procedure MUST return one result set as follows.

3.1.5.57.1 Account Status Result Set

The Account Status Result Set returns account information for the non-deleted Users matching the specified login name or email address. This parameter MUST be returned and MUST contain one row for each User (each principal (1) which is not a domain group) in the UserInfo Table (section 2.2.7.10) who is not marked as deleted, and matches either the login name specified by the @Login parameter or the email address specified by the @Email parameter. The Account Status Result Set MUST contain no rows if no matching Users are found.

```
      tp_Login                      nvarchar(255),
      tp_Email                      nvarchar(255),
      tp_SystemID                   varbinary(512);
```

tp_Login: The login name of the matching User.

tp_Email: The email address of the matching User.

tp_SystemID: The **SystemID** (section 2.2.1.12) of the matching User.

Copyright © 2014 Microsoft Corporation.

3.1.5.58 proc_SecGetAclFromScope

The **proc_SecGetAclFromScope** stored procedure is invoked to obtain the ACL and the rights granted to an anonymous user for a particular scope.

```
PROCEDURE proc_SecGetAclFromScope(
      @SiteId                       uniqueidentifier,
      @ScopeId                      uniqueidentifier
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) for the site collection containing the scope.

@ScopeId: The Scope Identifier (section 2.2.1.8) for the scope for which the ACL and anonymous user permissions are requested.

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

The **proc_SecGetAclFromScope** stored procedure MUST return a single result set as defined below:

3.1.5.58.1 ACL and Permission Result Set

The ACL and Permission Result Set (section 2.2.5.1) contains permission information for the specified scope. If either of the input parameters is not valid, the ACL and Permission Result Set MUST contain zero rows; otherwise, one row MUST be returned. The ACL and Permission Result Set is defined in Common Result Sets ACL and Permission Result Set section.

3.1.5.59 proc_SecGetAllAclsForSite

The **proc_SecGetAllAclsForSite** stored procedure is invoked to list all Scope Identifiers (section 2.2.1.8) and their associated ACL and anonymous permission masks in a site collection.

```
PROCEDURE proc_SecGetAllAclsForSite(
      @SiteId                       uniqueidentifier,
      @MaxCount                     int,
      @RowCount                     int               out
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection for which the information is requested.

@MaxCount: The maximum size, in rows, requested for the result set. If the number of unique scopes within the site collection exceeds **@MaxCount**, the Access Control List Result Set (section 3.1.5.59.1) will not be returned, otherwise an empty result set will be returned.

@RowCount: Output parameter indicating the number of unique scopes that exist in the site collection.

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

If the total count of unique Access Control Lists in the site collection does not exceed the value of the input parameter **@MaxCount**, then the **proc_SecGetAllAclsForSite** stored procedure MUST

Copyright © 2014 Microsoft Corporation.

return the Access Control List Result Set. Otherwise, the **proc_SecGetAllAclsForSite** stored procedure MUST return an Empty Result Set (section 2.2.5.8).

3.1.5.59.1 Access Control List Result Set

The Access Control List Result Set returns one row for each unique scope and associated Access Control List defined in the site collection.

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

ScopeId: The Scope Identifier (section 2.2.1.8) of the scope.

Acl: The access control list (ACL) associated with the scope.

AnonymousPermMask: The WSS Rights Mask (section 2.2.2.13) for anonymous users for this scope.

3.1.5.60 proc_SecGetAllGroupsAndMembershipInfo

The **proc_SecGetAllGroupsAndMembershipInfo** stored procedure is invoked to return information about all site groups and site group members within a site collection.

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

@SiteID: The Site Collection Identifier (section 2.2.1.9) of the site collection to find all site groups in.

Return Code Values: The **proc_SecGetAllGroupsAndMembershipInfo** stored procedure returns an integer which MUST be 0.

The **proc_SecGetAllGroupsAndMembershipInfo** stored procedure MUST return two result sets in the following order**:**

3.1.5.60.1 Groups Result Set

The Groups Result Set returns all site groups within the site collection, one row per site group, sorted by Site Group Identifier (section 2.2.1.9), in ascending order. The Groups Result Set MUST be empty if there is no site group on the site collection.

```
ID                      int,
Title                   nvarchar(255),
Description             nvarchar(512),
Owner                   int,
OwnerIsUser             bit;
```

ID: The Site Group Identifier of the site group.

Title: The title of the site group.

Copyright © 2014 Microsoft Corporation.

Description: The description of the site group.

Owner: The User Identifier (section 2.2.1.13) or Site Group Identifier of the owner of the site group.

OwnerIsUser: A bit value indicating whether the owner of the group is a principal (1) or a site group. If the owner is a principal (1), this parameter's value MUST be 1, otherwise, its value MUST be 0.

3.1.5.60.2 Group Membership Result Set

The Group Membership Result Set returns information about site group members. A member can be a User or a domain group. Every row represents one site group member. The first column is the GroupId, the Site Group Identifier (section 2.2.1.10), followed by columns about the site group member. The Group Membership Result Set is sorted by GroupId, in ascending order. If a member belongs to more than one site group, the member's information will appear in multiple rows, with different GroupIds. The Group Membership Result Set can be empty if none of the site groups have any members.

```
GroupId                 int,
tp_Id                   int,
tp_SystemID             varbinary(512),
tp_Title                nvarchar(255),
tp_Login                nvarchar(255),
tp_Email                nvarchar(255),
tp_Notes                nvarchar(1023),
tp_SiteAdmin            bit,
tp_DomainGroup          bit;
```

GroupId: The Site Group Identifier of the site group which contains this member.

tp_Id: The user identifier of the member who belongs to the site group specified by the GroupId in the first column.

tp_SystemID: The **SystemID** (section 2.2.1.12) of the member.

tp_Title: The display name of the member.

tp_Login: The login name of the member.

tp_Email: The email address of the member. This parameter can be empty, but it MUST NOT be NULL.

tp_Notes: A string which contains extra information about the member. It can be empty, but it MUST NOT be NULL.

tp_SiteAdmin: A bit value indicating whether the member is a site collection Administrator. If the member is a site collection Administrator, this parameter's value MUST be 1, otherwise, it MUST be 0.

tp_DomainGroup: A bit value indicating whether the member is a domain group. If the site group member is a domain group, this parameter's value MUST be 1. Otherwise, its value MUST be 0.

Copyright © 2014 Microsoft Corporation.

3.1.5.61 proc_SecGetCompleteWebRoleMemberList

The **proc_SecGetCompleteWebRoleMemberList** stored procedure is invoked to list all role assignments for all the principals with permissions on a specified site (2).

```
PROCEDURE proc_SecGetCompleteWebRoleMemberList(
      @SiteId                       uniqueidentifier,
      @WebId                        uniqueidentifier,
      @LatestSecurityVersion        bigint            OUTPUT
);
```

@SiteId: Site Collection Identifier (section 2.2.1.9) of the site collection containing the Site whose permission information is requested.

@WebId: Site Identifier (section 2.2.1.11) of the Site whose permission information is requested.

@LatestSecurityVersion: The current security version value for the specified site collection. This value is incremented with every security setting modification on the site collection.

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

The **proc_SecGetCompleteWebRoleMemberList** stored procedure MUST return one result set.

3.1.5.61.1 Role Member Result Set

The Role Member Result Set returns one row for each principal's role assignment in effect for the specified Site. If the **@SiteId** or **@WebId** parameters are not valid, zero rows will be returned.

```
      RoleId                        int,
      tp_Id                         int,
      tp_SystemID                   varbinary(512),
      tp_DomainGroup                bit;
```

RoleId: The Role Identifier (section 2.2.1.7) of the role for the role assignment.

tp_Id: The user identifier of a principal (1).

tp_SystemID: The **SystemID** (section 2.2.1.12) of the principal (1)

tp_DomainGroup: Set to 1 if the principal (1) is a domain group; otherwise, 0.

3.1.5.62 proc_SecGetCurrentUsersCount

The **proc_SecGetCurrentUsersCount** stored procedure returns a result set containing a count of users in the specified site collection.

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

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection.

Copyright © 2014 Microsoft Corporation.

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

The **proc_SecGetCurrentUsersCount** stored procedure MUST return a single result set for the specified site collection as follows.

3.1.5.62.1 User Count Result Set

The User Count Result Set returns the number of Users registered with this site collection when configured in Active Directory Creation mode. When not in Active Directory mode, UsersCount will always be 1. The User Count Result Set MUST return one row for a given valid **@SiteId**, otherwise if **@**SiteId is invalid, zero rows are returned.

```
UsersCount                  int,
UserQuota                   int,
{StorageQuotaError}         int;
```

UsersCount: Contains the integer value for the number of Users registered with the specified site collection when configured in Active Directory Creation mode. In any other configuration, **UsersCount** MUST return 1.

UserQuota: Contains the limit for the number of Users allowed in the specified site collection. A value of 0 specifies no limit on the number of allowed Users.

{StorageQuotaError}: An error number generated when a site collection is over quota or locked. Its value MUST be listed in the following table.

Value	Description
0	Default value, no error.
212	The site collection is locked.
1816	Quota has been exceeded on this site collection.

3.1.5.63 proc_SecGetDomainGroupMapData

The **proc_SecGetDomainGroupMapData** stored procedure is invoked to retrieve the domain group map cache information for a site collection.

```
PROCEDURE proc_SecGetDomainGroupMapData(
      @SiteId                       uniqueidentifier,
      @DGCacheVersion               bigint
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) for the site collection containing the site.

@DGCacheVersion: The version number of the Domain Group Map Cache in the WFE. This can be compared with the Domain Group Map Cache version number on the BEDS returned in the Domain Group Cache Versions Result Set (section 2.2.5.4) to determine whether updates are needed. A **@DGCacheVersion** value of -2 specifies that information about the Domain Group Map Cache is not requested, and the Domain Group Cache BEDS Update Result Set (section 2.2.5.3) and the Domain Group Cache WFE Update Result Set (section 2.2.5.5) MUST NOT be returned.

Copyright © 2014 Microsoft Corporation.

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

The **proc_SecGetDomainGroupMapData** stored procedure MUST return one or two result sets in the order specified.

3.1.5.63.1 Domain Group Cache Versions Result Set

The Domain Group Cache Versions Result Set contains information about the version numbers associated with the Domain Group Map Caches on the WFE and on the BEDS for the specified site collection.

The Domain Group Cache Versions Result Set MUST be returned and MUST contain one row of version number data. If the specified @DGCacheVersion value is -2, then all columns returned MUST have the value -2, indicating that the value MUST NOT be used for comparison. The Domain Group Cache Versions Result Set is defined in the Common Result Sets Domain Group Cache Versions Result Set (section 2.2.5.4).

3.1.5.63.2 Domain Group Cache BEDS Update Result Set

The Domain Group Cache BEDS Update Result Set contains information to be used in re-computing the Domain Group map cache, which contains the mapping of domain groups to the site groups that they are members of. The presence of the Domain Group Cache BEDS Update Result Set means the database's copy of the Domain Group map cache is out of date and MUST be recomputed to ensure that proper security checks can be made.

The Domain Group Cache BEDS Update Result Set MUST be returned if @DGCacheVersion does not equal "-2" and the value of RealVersion is greater than the value of CachedVersion in the results of the Domain Group Cache Versions Result Set (section 2.2.5.4). Otherwise, the Domain Group Cache BEDS Update Result Set MUST NOT be returned.

If the Domain Group Cache BEDS Update Result Set is returned, there MUST be one row in the Domain Group Cache BEDS Update Result Set for each domain group which is a member of a site group in the site collection, ordered by the identifier of the domain groups. A row is also returned for every other domain group that does not have a GroupId tied to it, including domain groups with NULL GroupId.

The Domain Group Cache BEDS Update Result Set is defined in the Common Result Sets Domain Group Cache BEDS Update Result Set (section 2.2.5.3).

3.1.5.63.3 Domain Group Cache WFE Update Result Set

The Domain Group Cache WFE Update Result Set contains the binary data needed to refresh the Domain Group map cache, as defined in the WSS External Group Map Cache Format (section 2.2.4.7). If the Domain Group Cache WFE Update Result Set is returned, it indicates that the BEDS Domain Group map cache is up to date, and the WFE cache can be refreshed if necessary.

The Domain Group Cache WFE Update Result Set MUST be returned if @DGCacheVersion does not equal "-2" and the value of RealVersion is less than or equal to the value of CachedVersion in the results of the Domain Group Cache Versions Result Set (section 2.2.5.4). Otherwise, the Domain Group Cache WFE Update Result Set MUST NOT be returned.

If the Domain Group Cache WFE Update Result Set is returned, it MUST contain one row.

The Domain Group Cache WFE Update Result Set is defined in the Common Result Sets Domain Group Cache WFE Update Result Set (section 2.2.5.5).

Copyright © 2014 Microsoft Corporation.

3.1.5.64 proc_SecGetGroupById

The **proc_SecGetGroupById** stored procedure is invoked to check whether the specified Group (either a site group or a domain group) exists in the specified site collection.

```
PROCEDURE proc_SecGetGroupById(
      @SiteId                       uniqueidentifier,
      @GroupId                      int,
      @Count                        int               OUTPUT
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection to search for the Group.

@GroupId: The identifier (a Site Group Identifier (section 2.2.1.10) for site groups, or a User Identifier (section 2.2.1.13) for domain groups) of the specified Group.

@Count: Specifies whether the specified Group exists in the site collection as either a site group or a domain group. If the specified Group exists, **@Count** MUST be 1; otherwise, **@Count** MUST be 0.

Return Code Values**:** The **proc_SecGetGroupById** stored procedure returns an integer which MUST be 0.

The **proc_SecGetGroupById** stored procedure MUST NOT return any result sets.

3.1.5.65 proc_SecGetGroupOwner

The **proc_SecGetGroupOwner** stored procedure is invoked to retrieve the User Identifier (section 2.2.1.13) or Site Group Identifier (section 2.2.1.10) for the owner of a site group.

```
PROCEDURE proc_SecGetGroupOwner(
      @SiteId                       uniqueidentifier,
      @GroupId                      int,
      @OwnerId                      int OUTPUT
);
```

@SiteId: Site Collection Identifier (section 2.2.1.9) for the site collection containing the site group.

@GroupId: The site group's identifier.

@OwnerId: An output parameter containing the User Identifier or Site Group Identifier of the site group owner, or "-1" if the specified site group does not exist.

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

Value	Description
0	Successful execution.
1319	The site group does not exist.

The **proc_SecGetGroupOwner** stored procedure MUST NOT return a result set.

Copyright © 2014 Microsoft Corporation.

3.1.5.66 proc_SecGetGroupSecurityScopes

The **proc_SecGetGroupSecurityScopes** stored procedure is invoked to retrieve a site group's role assignments information on all security scopes within a given site collection.

```
PROCEDURE proc_SecGetGroupSecurityScopes(
      @SiteId                       uniqueidentifier,
      @PrincipalId                  int
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the site group.

@PrincipalId: The identifier of the site group.

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

The **proc_SecGetGroupSecurityScopes** stored procedure MUST return one result set.

3.1.5.66.1 Security Scopes Result Set

The Security Scopes Result Set returns scope information for the specified security principal (2). The Security Scopes Result Set will always be returned with zero or more rows.

```
      ScopeUrl                      nvarchar(260),
      Title                         nvarchar(255);
```

ScopeUrl: The URL to the root of the Security Scope. This value can be an empty string if the scope is applied to the entire site collection.

Title: The name of the role to which the site group is assigned on the Security Scope specified by **{ScopeUrl}**.

3.1.5.67 proc_SecGetIndividualUrlSecurityCheckEventReceivers

The **proc_SecGetIndividualUrlSecurityCheckEventReceivers** stored procedure is invoked to request security information and event receivers information about a document at a specified location. If the document does not exist, **proc_SecGetIndividualUrlSecurityCheckEventReceivers** provides information about the specified location.

```
PROCEDURE proc_SecGetIndividualUrlSecurityCheckEventReceivers(
      @SiteId                       uniqueidentifier,
      @WebId                        uniqueidentifier,
      @FullUrl                      nvarchar(260),
      @DirName                      nvarchar(256),
      @LeafName                     nvarchar(128),
      @UserId                       int,
      @AttachmentsFlag              tinyint,
      @bGetAttachmentWritePerm      bit,
      @bGetListMetaData             bit             = 0,
      @bGetListScopes               bit             = 0,
      @bLockLists                   bit             = 0,
      @Level                        tinyint         = NULL,
```


Copyright © 2014 Microsoft Corporation.

```
        @HasEventReceiver               bit                 OUTPUT
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the document location.

@WebId: The Site Identifier (section 2.2.1.11) of the site containing the document location.

@FullUrl: The store-relative form URL of the document location, used to find a containing list (1) or document library if the document does not exist.

@DirName: The directory name of the document location.

@LeafName: The leaf name of the document location.

@UserId: The User Identifier (section 2.2.1.13) of the current user, used to check for access privileges.

@AttachmentsFlag: An Attachments Flag (section 2.2.2.1) value specifying whether the document location is, or is contained within, an attachments folder.

@bGetAttachmentWritePerm: A bit flag specifying whether to return information about the write permissions of the current user to save an attachment at the specified document location. If this parameter is set to 1, and the **@AttachmentsFlag** parameter is not 0 or NULL, **proc_SecGetIndividualUrlSecurityCheckEventReceivers** MUST return the information about the Current User's permissions as part of the Individual URL Security Result Set (section 2.2.5.10). This parameter MUST be ignored if **@AttachmentsFlag** is 0 or NULL.

@bGetListMetaData: A bit flag specifying whether List metadata is requested for the List or document library containing the document location.

@bGetListScopes: A bit flag specifying whether scope information is requested for the List or document library containing the document location.

@bLockLists: A bit flag specifying whether to prevent updates to the table of Lists when obtaining the List Identifier (section 2.2.1.5) of the List or document library containing the document location. If this parameter is set to 1, the table MUST be locked while obtaining the List Identifier.

@Level: Specifies the Publishing Level Type (section 2.2.2.6) value to return in the Individual URL Security Result Set. If this parameter is NULL, the publishing level of the current version of the document for the current user MUST be used.

@HasEventReceiver: An output parameter that **proc_SecGetIndividualUrlSecurityCheckEventReceivers** MUST set to 1 if at least one event receiver is registered for the specified document; otherwise 0.

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

The **proc_SecGetIndividualUrlSecurityCheckEventReceivers** stored procedure MUST return between 1 and 4 result sets, in the order listed in the following.

Copyright © 2014 Microsoft Corporation.

3.1.5.67.1 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 or document library, security information is returned from the effective 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 or document library. Otherwise, the NULL Individual URL Security Result Set (section 2.2.5.14) MUST be returned instead. If returned, the Individual URL Security Result Set MUST contain a single row.

The Individual URL Security Result Set is defined in the Common Result Sets Individual URL Security Result Set (section 2.2.5.10).

3.1.5.67.2 NULL Individual URL Security Result Set

The NULL Individual URL Security Result Set indicates that the document location is not contained within a List or document library. The NULL Individual URL Security Result Set MUST only be returned if the specified document location is not contained within a List or document library.

The NULL Individual URL Security Result Set is defined in the Common Result Sets NULL Individual URL Security Result Set (section 2.2.5.14).

3.1.5.67.3 List Metadata Result Set

The List Metadata Result Set returns the metadata for the List or document library containing the specified document location. The List Metadata Result Set MUST only be returned if the @bGetListMetadata parameter is set to 1 and the document location is contained within a List or document library. If returned, the List Metadata Result Set MUST contain a single row.

The List Metadata Result Set is defined in the Common Result Sets List Metadata Result Set (section 2.2.5.12).

3.1.5.67.4 Event Receivers Result Set

The Event Receivers Result Set contains information about event receivers defined for the containing List or document library. This result set MUST only be returned if the @bGetListMetaData parameter is set to 1 and the document location is contained within a List or document library. If returned, the Event Receivers Result Set MUST contain one row for each event receiver registered for the List or document library.

The Event Receivers Result Set is defined in the Common Result Sets Event Receivers Result Set (section 2.2.5.9).

3.1.5.67.5 List Scopes Result Set

The List Scopes Result Set returns the security information for the scope associated with the document location. The List Scopes Result Set MUST only be returned if the @bGetListScopes parameter is set to 1 and the document location is within a list (1) or document library. If returned, the List Scopes Result Set MUST contain one row.

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


Copyright © 2014 Microsoft Corporation.

ScopeId: The Scope Identifier (section 2.2.1.8) for the scope associated with the document location.

Acl: The WSS ACL (section 2.2.4.6) of the scope associated with the document location.

AnonymousPermMask: The WSS Rights Mask (section 2.2.2.13) that applies to an anonymous user, or a user with no assigned rights, in the scope associated with the document location.

3.1.5.68 proc_SecGetPrincipalByEmail

The **proc_SecGetPrincipalByEmail** stored procedure is invoked to return user information by an email address.

```
PROCEDURE proc_SecGetPrincipalByEmail(
      @SiteId                       uniqueidentifier,
      @Email                        nvarchar(255)
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) for the site collection containing the user.

@Email: The user's email address. If this parameter is NULL, the email address MUST be set to the empty string.

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

The **proc_SecGetPrincipalByEmail** stored procedure MUST return a single result set.

3.1.5.68.1 Principal User Information Result Set

The Principal User Information Result Set returns information about the User associated with the specified email address. The Principal User Information Result Set MUST contain zero rows if no Users have the specified email address or if the associated User has been marked as deleted. The Principal User Information Result Set is defined in the Common Result Sets Principal User Information Result Set (section 2.2.5.17).

3.1.5.69 proc_SecGetPrincipalById

The **proc_SecGetPrincipalById** stored procedure is invoked to return information about a principal (1) or collection of principals based on a specified site group identifier or user identifier.

```
PROCEDURE proc_SecGetPrincipalById(
      @SiteId                       uniqueidentifier,
      @PrincipalId                  int,
      @GetSTSToken                  bit               = 0,
      @GetExternalToken             bit               = 0,
      @GetExpandedSTSGroup          bit               = 0
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the principal (1).

@PrincipalId: The site group identifier of the site group or user identifier of the principal (1) to return information for.

Copyright © 2014 Microsoft Corporation.

@GetSTSToken: If this parameter is set to 1, it indicates the **{tp_token}** value MUST be returned in both instances of the Principal User Information Result Set.

@GetExternalToken: If this parameter is set to 1, it indicates the **{tp_ExternalTokenLastUpdated}** and **{tp_ExternalToken}** values MUST be returned in both instances of the Principal User Information Result Set.

@GetExpandedSTSGroup: If this parameter is set to 1, it indicates a second Principal User Information Result Set MUST be returned when the first instance of the Principal User Information Result Set has zero rows. The second result set contains information about group membership for the site group represented by **@PrincipalId**. The rows returned in the second result set are all the members of that site group.

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

The **proc_SecGetPrincipalById** stored procedure MUST return one result set which can be returned either once or twice as described in the following.

3.1.5.69.1 User Information Result Set

The User Information Result Set returns information on a specified principal (1) or the members of a site group.

If **@PrincipalId** matches a principal (1), the User Information Result Set MUST be returned once with a single row of data for the principal.

Otherwise, the User Information Result Set MUST be returned with zero rows, and if **@GetExpandedSTSGroup** is 1, another instance of the User Information Result Set MUST be returned, with one row of data for each principal (1) within the site group.

```
tp_Id                          int,
tp_SystemID                    varbinary(512),
tp_Title                       nvarchar(255),
tp_Login                       nvarchar(255),
tp_Email                       nvarchar(255),
tp_Notes                       nvarchar(1023),
tp_SiteAdmin                   bit,
tp_DomainGroup                 bit,
{tp_ExternalTokenLastUpdated}  datetime,
{tp_ExternalToken}             image,
{tp_Token}                     image;
```

tp_Id: The user identifier of the principal (1).

tp_SystemID: The principal's **SystemId** (section 2.2.1.12).

tp_Title: The principal's display name.

tp_Login: The principal's login name.

tp_Email: The principal's email address.

tp_Notes: A longer descriptive text associated with the principal (1).

tp_SiteAdmin: Set to 1 if the principal (1) is a site administrator; otherwise, 0.

Copyright © 2014 Microsoft Corporation.

tp_DomainGroup: Set to 1 if the principal (1) is a domain group; otherwise, 0.

{Tp_ExternalTokenLastUpdated}: A timestamp in UTC format specifying the time when the External Group Token (section 2.2.4.2) was last updated. This parameter MUST be NULL if **@GetExternalToken** is 0.

{Tp_ExternalToken}: An External Group Token value specifying the domain group membership of the principal (1). This parameter MUST be NULL if **@GetExternalToken** is 0.

{Tp_Token}: A WSS User Token (section 2.2.4.9) value specifying the site group membership of the principal (1). This parameter MUST be NULL if **@GetSTSToken** is 0.

3.1.5.70 proc_SecGetPrincipalByLogin

The **proc_SecGetPrincipalByLogin** stored procedure is invoked to return security and attribute information for a principal (1) (a user or domain group) identified by a specified login name.

```
PROCEDURE proc_SecGetPrincipalByLogin(
      @SiteId                     uniqueidentifier,
      @Login                      nvarchar(255),
      @GetSTSToken                bit               = 0,
      @GetExternalToken           bit               = 0
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) for the site collection associated with the principal (1) whose information is requested.

@Login: The login name of the principal (1). This parameter MUST NOT be NULL.

@GetSTSToken: A bit flag specifying whether to include site group membership information for the principal (1). If this parameter is set to 1, a WSS User Token (section 2.2.4.9) containing the principal's site group membership information MUST be returned in the User Information Result Set.

@GetExternalToken: A bit flag specifying whether to include domain group membership and related timestamp information for the principal (1). If this parameter is set to 1, an External Group Token (section 2.2.4.2) containing the principal's domain group membership information and a timestamp with its most recent update time MUST be returned in the User Information Result Set.

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

The **proc_SecGetPrincipalByLogin** stored procedure MUST return a single result set as follows.

3.1.5.70.1 User Information Result Set

The User Information Result Set returns security and attribute information about a specified principal (1). The User Information Result Set MUST be returned, and MUST contain one row if the @Login parameter matches an existing principal who is not marked as deleted in the UserInfo Table (section 2.2.7.10); otherwise it MUST contain no rows.

```
      tp_Id                       int,
      tp_SystemID                 varbinary(512),
      tp_Title                    nvarchar(255),
      tp_Login                    nvarchar(255),
      tp_Email                    nvarchar(255),
      tp_Notes                    nvarchar(1023),
```


Copyright © 2014 Microsoft Corporation.

```
    tp_SiteAdmin                    bit,
    tp_DomainGroup                  bit,
    {tp_ExternalTokenLastUpdated}   datetime,
    {tp_ExternalToken}              image,
    {tp_Token}                      image;
```

tp_Id: The User Identifier (section 2.2.1.13) for the specified principal (1).

tp_SystemID: The **SystemID** (section 2.2.1.12) for the principal (1).

tp_Title: The display name of the principal (1).

tp_Login: The principal's login name.

tp_Email: The principal's email address.

tp_Notes: A descriptive text string associated with the principal (1).

tp_SiteAdmin: A bit set to 1 if the principal (1) has administrator rights in the site collection; otherwise, 0.

tp_DomainGroup: A bit set to 1 if the principal (1) is a domain group; otherwise 0, indicating the principal is a user.

{tp_ExternalTokenLastUpdated}: A datetime in UTC format specifying the time when the external group token was last updated. This value MUST be NULL if the **@GetExternalToken** parameter is not set to 1.

{tp_ExternalToken}: An External Group Token (section 2.2.4.2) value specifying the domain group membership of the principal (1). This value MUST be NULL if the **@GetExternalToken** parameter is not set to 1.

{tp_Token}: A WSS user token value specifying the site group membership of the principal (1). This value MUST be NULL if the **@GetSTSToken** parameter is not set to 1.

3.1.5.71 proc_SecGetPrincipalByLogin20

The **proc_SecGetPrincipalByLogin20** stored procedure is invoked to return security principal (2) information based on up to 20 separate login names.

```
PROCEDURE proc_SecGetPrincipalByLogin20(
      @SiteId                       uniqueidentifier,
      @PrincipalId01                nvarchar(255),
      @PrincipalId02                nvarchar(255),
      @PrincipalId03                nvarchar(255),
      @PrincipalId04                nvarchar(255),
      @PrincipalId05                nvarchar(255),
      @PrincipalId06                nvarchar(255),
      @PrincipalId07                nvarchar(255),
      @PrincipalId08                nvarchar(255),
      @PrincipalId09                nvarchar(255),
      @PrincipalId10                nvarchar(255),
      @PrincipalId11                nvarchar(255),
      @PrincipalId12                nvarchar(255),
      @PrincipalId13                nvarchar(255),
      @PrincipalId14                nvarchar(255),
```


Copyright © 2014 Microsoft Corporation.

```
      @PrincipalId15                nvarchar(255),
      @PrincipalId16                nvarchar(255),
      @PrincipalId17                nvarchar(255),
      @PrincipalId18                nvarchar(255),
      @PrincipalId19                nvarchar(255),
      @PrincipalId20                nvarchar(255)
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) for the site collection that contains the security principals (2).

@PrincipalId##: The login names for Users to be returned.

Return Values: The **proc_SecGetPrincipalByLogin20** stored procedure returns an integer return code which MUST be 0.

The **proc_SecGetPrincipalByLogin20** stored procedure MUST return one result set of security principal (2) information for each non-NULL **@**PrincipalId##:

3.1.5.71.1 User Information Result Set

The User Information Result Set returns User information on a specified security principal (2). The User Information Result Set MUST always return one row if the **@PrincipalId##** matches the ID of a security principal (2). The User Information Result Set is defined in the **proc_SecGetPrincipalByLogin** User Information Result Set (section 3.1.5.15.5).

3.1.5.72 proc_SecGetPrincipalDisplayInformation20

The **proc_SecGetPrincipalDisplayInformation20** stored procedure is invoked to return security principal (2) or site group information for up to 20 principal identifiers.

```
PROCEDURE proc_SecGetPrincipalDisplayInformation20(
      @SiteId                       uniqueidentifier,
      @WebId                        uniqueidentifier,
      @PrincipalId01                int,
      @PrincipalId02                int,
      @PrincipalId03                int,
      @PrincipalId04                int,
      @PrincipalId05                int,
      @PrincipalId06                int,
      @PrincipalId07                int,
      @PrincipalId08                int,
      @PrincipalId09                int,
      @PrincipalId10                int,
      @PrincipalId11                int,
      @PrincipalId12                int,
      @PrincipalId13                int,
      @PrincipalId14                int,
      @PrincipalId15                int,
      @PrincipalId16                int,
      @PrincipalId17                int,
      @PrincipalId18                int,
      @PrincipalId19                int,
      @PrincipalId20                int
);
```


Copyright © 2014 Microsoft Corporation.

@SiteId: The Site Collection Identifier (section 2.2.1.9) for the site collection containing the security principals (2) and the Security Groups to be listed.

@WebId: A Site Identifier (section 2.2.1.11) for a site. This parameter is ignored.

@PrincipalId##: The identifier for a security principal (2) or a site group to be returned. Result sets are returned for each non-NULL @PrincipalId## parameter.

Return Values: The **proc_SecGetPrincipalDisplayInformation20** stored procedure returns an integer return code which MUST be 0.

The **proc_SecGetPrincipalDisplayInformation20** stored procedure MUST return one or two result sets for each non-NULL **@PrincipalId##**.

3.1.5.72.1 Principal Display Information Result Set

The Principal Display Information Result Set Returns information on a specified **@PrincipalId##**. The **@PrincipalId##** MUST match the ID of either a site group or a security principal (2). If **@PrincipalId##** matches the ID of a site group, the Principal Display Information Result Set MUST always return one row. If the **@PrincipalId##** matches the ID of a security principal (2), **proc_SecGetPrincipalDisplayInformation20** MUST return two result sets**:** the first MUST have no rows and the second MUST have one row for the matching security principal (2).

```
IsUser                      bit,
IsSiteGroup                 bit,
UserID                      int,
UserSID                     varbinary(512),
UserName                    nvarchar(255),
UserLogin                   nvarchar(255),
UserEmail                   nvarchar(255),
UserNotes                   nvarchar(1023),
UserSiteAdmin               bit,
UserDomainGroup             bit,
GroupID                     int,
GroupName                   nvarchar(255),
GroupDescription            nvarchar(512),
GroupOwnerID                int,
GroupOwnerIsUser            bit,
GroupType                   tinyint;
```

IsUser: 1 if this row represents a security principal (2), otherwise 0.

IsSiteGroup: 1 if this row represents a site group, otherwise 0.

UserID: The security principal (2) identifier. MUST be NULL if this row represents a site group. MUST match **@PrincipalId##** if this row represents a security principal (2).

UserSID: The **SystemID** (section 2.2.1.12) of the security principal (2). MUST be NULL if this row represents a site group.

UserName: The user's name as used for display. MUST be NULL if this row represents a site group.

Copyright © 2014 Microsoft Corporation.

UserLogin: The user's login name, for example, EXAMPLE\username. MUST be NULL if this row represents a site group.

UserEmail: The user's email address, for example, username@mail.example.com. MUST be NULL if this row represents a site group.

UserNotes: A longer description text associated with the security principal (2). MUST be NULL if this row represents a site group.

UserSiteAdmin: 1 if the security principal (2) is a site collection administrator, otherwise 0. MUST be NULL if this row represents a Security Group.

UserDomainGroup: 1 if the security principal (2) is a domain group, otherwise 0. MUST be NULL if this row represents a site group.

GroupID: The ID of the Security Group. MUST be NULL if this row represents a security principal (2). MUST match **@PrincipalId##** if this row represents a site group.

GroupName: The name of the site group as used for display. MUST be NULL if this row represents a security principal (2).

GroupDescription: The description of the site group. MUST be NULL if this row represents a security principal (2).

GroupOwnerID: The identifier of the owner of the site group. MUST be NULL if this row represents a security principal (2).

GroupOwnerIsUser: 1 if the owner of the site group is a security principal (2). MUST be NULL if this row represents a security principal (2).

GroupType: 0 if this row represents a site group. This parameter MUST be NULL if this row represents a security principal (2).

3.1.5.73 proc_SecGetRoleAssignments

The **proc_SecGetRoleAssignments** stored procedure is invoked to request the WSS Access Control Entries (section 2.2.4.5) for a specified scope in a site collection.

```
PROCEDURE proc_SecGetRoleAssignments (
      @SiteId                       uniqueidentifier,
      @ScopeId                      uniqueidentifier
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the requested scope.

@ScopeId: The Scope Identifier (section 2.2.1.8) of the scope containing the requested ACEs.

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

The **proc_SecGetRoleAssignments** stored procedure MUST return one result set. It will return zero or more rows.

Copyright © 2014 Microsoft Corporation.

3.1.5.73.1 WSSACE Result Set

The WSSACE Result Set MUST return one row for each role assignment matching the given scope within the site collection. Each row represents a WSS ACE (section 2.2.4.5).

```
PrincipalId                    int,
PermMask                       bigint;
```

PrincipalId: A 4 byte signed integer that specifies the identifier of the WSS security principal (2) for this access control entry (ACE).

PermMask: A WSS Rights Mask (section 2.2.2.13) that contains the list of rights that is granted to the WSS security principal (2).

3.1.5.74 proc_SecGetRoleBindingsForAllPrincipals

The **proc_SecGetRoleBindingsForAllPrincipals** stored procedure is invoked to list the role assignments for all principals in a scope.

```
PROCEDURE proc_SecGetRoleBindingsForAllPrincipals(
      @SiteId                      uniqueidentifier,
      @ScopeId                     uniqueidentifier
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the scope.

@ScopeId: The Scope Identifier (section 2.2.1.8) of the scope containing the role assignments.

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

The **proc_SecGetRoleBindingsForAllPrincipals** stored procedure MUST return one result set as follows**:**

3.1.5.74.1 Role Assignment Result Set

The Role Assignment Result Set MUST return one row for each undeleted principal (1) in the scope.

```
RoleId                         int,
PrincipalId                    int;
```

RoleId: The Role Identifier (section 2.2.1.7) of a role associated with the role assignment.

PrincipalId: The User Identifier (section 2.2.1.13) of the principal (1).

3.1.5.75 proc_SecGetRoleDefs

proc_SecGetRoleDefs is invoked to retrieve role definition information for items in the specified site collection and scope.

```
PROCEDURE proc_SecGetRoleDefs(
      @SiteId                      uniqueidentifier,
      @ScopeId                     uniqueidentifier
```


Copyright © 2014 Microsoft Corporation.

```
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) for the desired site collection. This parameter MUST correspond to a valid site collection.

@ScopeId: The Scope Identifier (section 2.2.1.8) of the desired scope.

Return Values: The proc_SecGetRoleDefs stored procedure MUST return an integer return code of 0, and MUST return a role definition result set.

3.1.5.75.1 Role Definition Result Set

The Role Definition Result Set contains a list of role definitions. The Role Definition Result Set MUST contain one row for each role defined in the specified site collection at the specified scope.

```
RoleId                  int,
Title                   nvarchar(255),
Description             nvarchar(512),
Hidden                  bit,
RoleOrder               int,
Type                    tinyint,
WebId                   uniqueidentifier,
{PermMaskUpper}         int,
{PermMaskLower}         int,
WebGroupId              int;
```

RoleId: The Role Identifier (section 2.2.1.7) for the role.

Title: The display name of the role.

Description: A user-defined description of the role.

Hidden: A bit flag specifying whether the role definition MUST NOT be displayed in the WFE. When this bit is set to "1" the role definition MUST NOT be displayed. When this bit is set to "0" the role definition MUST be displayed.

RoleOrder: Specifies the order in which roles MUST be displayed in the WFE. Roles MUST be sorted first in ascending RoleOrder, then by descending Type.

Type: The Role Definition Type (section 2.2.3.16) of the role.

WebId: The Site Identifier (section 2.2.1.11) for the site within the site collection to which the role is assigned.

{PermMaskUpper}: An integer containing the high-order 32 bits of a WSS Rights Mask defining the permissions for the role.

{PermMaskLower}: An integer containing the low-order 32 bits of a WSS Rights Mask defining the permissions for the role.

WebGroupId: If the role was upgraded from a site group, the identifier of the site group from which it was upgraded. Otherwise, -1. If the role was upgraded from a site group, Type MUST be the identifier of the site group from which it was upgraded, otherwise it MUST be -1.

Copyright © 2014 Microsoft Corporation.

3.1.5.76 proc_SecGetRoleDefsForPrincipal

The **proc_SecGetRoleDefsForPrincipal** stored procedure returns information about the roles assigned to a principal (1) within a scope.

```
PROCEDURE proc_SecGetRoleDefsForPrincipal(
      @SiteId                       uniqueidentifier,
      @ScopeId                      uniqueidentifier,
      @PrincipalId                  int
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection that contains the Security Scope specified by **@ScopeId**.

@ScopeId: The Scope Identifier (section 2.2.1.11) of the Scope containing the role definition(s).

@PrincipalId: The User Identifier (section 2.2.1.13) of the principal (1).

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

The **proc_SecGetRoleDefsForPrincipal** stored procedure MUST return one result set.

3.1.5.76.1 Role Definition Assignment Result Set

The Role Definition Assignment Result Set returns information about each role that the principal (1) is assigned on the given scope. The Role Definition Assignment Result Set will contain one row for each role assignment defined within the scope specified by **@**ScopeId.

```
      RoleId                        int,
      Title                         nvarchar(255),
      Description                   nvarchar(512),
      Hidden                        bit,
      Type                          tinyint,
      {PermMaskUpper}               int,
      {PermMaskLower}               int,
      PrincipalId                   int;
```

RoleId: The Role Identifier (section 2.2.1.7) for the role.

Title: The display name of the role.

Description: A user-defined description of the role.

Hidden: Set to 1 to indicate that the role definition is not displayed on the "Add Users" page.

Type: The Role Definition Type (section 2.2.3.16) of role.

{PermMaskUpper}: An integer containing the high-order 32 bits of a WSS Rights Mask defining the permissions for the role.

{PermMaskLower}: An integer containing the low-order 32 bits of a WSS Rights Mask defining the permissions for the role.

PrincipalId: An identifier for the security principal (2). This MUST be the same as **@PrincipalId**.

Copyright © 2014 Microsoft Corporation.

3.1.5.77 proc_SecGetSecurityInfo

proc_SecGetSecurityInfo retrieves security permissions information about a document. The document can be specified using its document identifier (section 2.2.1.2) or its URL.

```
PROCEDURE proc_SecGetSecurityInfo(
      @SiteId                       uniqueidentifier,
      @WebId                        uniqueidentifier,
      @Url                          nvarchar(260),
      @DocId                        uniqueidentifier
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the document.

@WebId: The Site Identifier (section 2.2.1.11) of the site (2) containing the document. This parameter MUST be ignored.

@Url: The store-relative form URL for the document. The **@Url** parameter MUST be NULL to specify the document with the **@DocId** parameter.

@DocId: The document identifier (2) of the document. The **@DocId** parameter MUST be ignored and MAY be NULL if **@Url** specifies the document.

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

Value	Description
0	Successful execution.
3	The Document was not found.

The **proc_SecGetSecurityInfo** stored procedure MUST return a single Security Information result set upon successful execution, or none if the document is not found.

3.1.5.77.1 Security Information Result Set

The Security Information Result Set returns information about the security permissions on the document. The Security Information Result Set MUST be returned only if the document exists, and MUST return one row of information.

```
      ScopeId                       uniqueidentifier,
      ScopeUrl                      nvarchar(260),
      {IsScope}                     bit,
      Acl                           image,
      AnonymousPermMask             bigint;
```

ScopeId: Contains the Scope Identifier (section 2.2.1.8) of the scope that contains the document.

ScopeUrl: The store-relative form URL of the scope that contains the document.

{IsScope}: A flag indicating that the document is the securable object at the root of the scope. This value MUST be set to 1 if ScopeUrl is the store-relative form URL of the document; otherwise the value MUST be 0.

Copyright © 2014 Microsoft Corporation.

Acl: The Access Control List of the scope that contains the document.

AnonymousPermMask: The WSS Rights Mask (section 2.2.2.13) in effect for anonymous users in the scope.

3.1.5.78 proc_SecGetSiteAdmins

The **proc_SecGetSiteAdmins** stored procedure is invoked to return a list of All WSS site administrators not marked as deleted.

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

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the Administrators.

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

The **proc_SecGetSiteAdmins** stored procedure MUST return one result set as follows.

3.1.5.78.1 Site Administrators Result Set

The Site Administrators Result Set returns a list of All WSS site collection administrators not marked as deleted. The Site Administrators Result Set MUST be returned. The Site Administrators Result Set MUST return one row per administrator.

```
      tp_Id                           int,
      tp_SystemID                     varbinary(512),
      tp_Title                        nvarchar(255),
      tp_Login                        nvarchar(255),
      tp_Email                        nvarchar(255),
      tp_Notes                        nvarchar(1023),
      tp_SiteAdmin                    bit,
      tp_DomainGroup                  bit;
```

tp_Id: The security principal (2) ID of the administrator.

tp_SystemID: The **SystemID** (section 2.2.1.12) of the administrator.

tp_Title: The administrator's name, used for display on the WFE.

tp_Login: The administrator's site login, for example, EXAMPLE\username.

tp_Email: The administrator's email address, for example, "username@mail.example.com".

tp_Notes: Notes associated with the administrator.

tp_SiteAdmin: Set to 1, indicating the security principal (2) is a site administrator.

tp_DomainGroup: Set to 1 if the administrator is a domain group; otherwise, 0.

Copyright © 2014 Microsoft Corporation.

3.1.5.79 proc_SecGetSiteGroupById

The **proc_SecGetSiteGroupById** stored procedure gets information about a site group given its ID.

```
PROCEDURE proc_SecGetSiteGroupById(
@SiteId          uniqueidentifier,
@GroupId  int
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the site group.

@GroupId: The ID of the site group.

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

The **proc_SecGetSiteGroupById** stored procedure MUST return a single result set.

3.1.5.79.1 Site Group Result Set

The Site Group Result Set returns information about a site group. The Site Group Result Set MUST return zero or one rows. The Site Group Result Set contains a single row where Sec_SiteGroupsView.SiteWebID MUST equal **@**SiteId and Sec_SiteGroupsView.ID MUST equal **@GroupID**. If there is no match, the Site Group Result Set will be empty.

The Site Group Result Set is defined using T-SQL syntax, in the Sec_SiteGroupsView (section 2.2.7.6) section.

3.1.5.80 proc_SecGetSiteGroupByTitle

The **proc_SecGetSiteGroupByTitle** stored procedure is invoked to get site group information for the site group with the specified user-friendly name.

```
PROCEDURE proc_SecGetSiteGroupByTitle(
      @SiteId                        uniqueidentifier,
      @Title                         nvarchar (255)
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the site group.

@Title: The user-friendly name of the site group.

Return Values: The **proc_SecGetSiteGroupByTitle** stored procedure MUST return an integer return code of 0 and it MUST return a single result set.

3.1.5.80.1 Site Group Information Result Set

The Site Group Information Result Set returns the available site group information for the specified site group. The Site Group Information Result Set MUST contain one row of site group information if the **@SiteID** parameter value and the **@Title** parameter value match an existing site group; otherwise, zero rows MUST be returned.

Copyright © 2014 Microsoft Corporation.

The Site Group Information Result Set schema is defined in the Sec_SiteGroupsView (section 2.2.7.6).

3.1.5.81 proc_SecGetSiteGroupByTitle20

The **proc_SecGetSiteGroupByTitle20** stored procedure provides information about site groups in bulk, as specified by a set of up to 20 site group titles.

```
PROCEDURE proc_SecGetSiteGroupByTitle20(
@SiteId  uniqueidentifier,
@PrincipalId01          nvarchar(255),
@PrincipalId02          nvarchar(255),
@PrincipalId03          nvarchar(255),
@PrincipalId04          nvarchar(255),
@PrincipalId05          nvarchar(255),
@PrincipalId06          nvarchar(255),
@PrincipalId07          nvarchar(255),
@PrincipalId08          nvarchar(255),
@PrincipalId09          nvarchar(255),
@PrincipalId10          nvarchar(255),
@PrincipalId11          nvarchar(255),
@PrincipalId12          nvarchar(255),
@PrincipalId13          nvarchar(255),
@PrincipalId14          nvarchar(255),
@PrincipalId15          nvarchar(255),
@PrincipalId16          nvarchar(255),
@PrincipalId17          nvarchar(255),
@PrincipalId18          nvarchar(255),
@PrincipalId19          nvarchar(255),
@PrincipalId20          nvarchar(255)
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) for the site collection containing the specified site groups.

@PrincipalId##: Twenty Site Group Title fields (01 - 20).

Return Values: The proc_SecGetSiteGroupByTitle20 stored procedure MUST return an integer return code of 0.

The **proc_SecGetSiteGroupByTitle20** stored procedure MUST return one result set for each NON-NULL **@PrincipalId##**.

3.1.5.81.1 Site Group Information Result Set

The Site Group Information Result Set returns once per each NON-NULL @PrincipalId##, and the Site Group Information Result Set MUST contain one row of site group information if the @SiteID parameter value and the @PrincipalId## parameter value match an existing site group; otherwise, zero rows MUST be returned.

The Site Group Information Result Set is defined using T-SQL syntax, as defined by the Sec_SiteGroupsView (section 2.2.7.6).

Copyright © 2014 Microsoft Corporation.

3.1.5.82 proc_SecGetUserAccountDirectoryPath

The **proc_SecGetUserAccountDirectoryPath** stored procedure is invoked to return the User Account Directory Path of a specified site collection.

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

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection.

Return Code Values**:** The **proc_SecGetUserAccountDirectoryPath** stored procedure returns an integer which MUST be 0.

The **proc_SecGetUserAccountDirectoryPath** stored procedure MUST return the following result set:

3.1.5.82.1 User Account Directory Path Result Set

The User Account Directory Path Result Set returns the User Account Directory Path for the site collection. The User Account Directory Path Result Set MUST contain one row when the specified site collection exists, and it MUST contain zero rows when the specified site collection does not exist.

```
UserAccountDirectoryPath     nvarchar(512)
```

UserAccountDirectoryPath: Contains the User Account Directory Path for the site collection.

3.1.5.83 proc_SecGetUserPermissionOnGroup

The **proc_SecGetUserPermissionOnGroup** stored procedure is invoked to determine what permissions the requesting user has within a specified site group.

```
PROCEDURE proc_SecGetUserPermissionOnGroup(
      @SiteId                       uniqueidentifier,
      @GroupId                      int,
      @UserId                       int,
      @BelongsToGroup               bit,
      @CanViewMembership            bit               OUTPUT,
      @CanEditMembership            bit               OUTPUT,
      @GroupOwnerId                 int               OUTPUT,
      @IsExplicitlyInMembership     bit               OUTPUT
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the specified group.

@GroupId: Identifier of the site group whose permissions are to be returned.

@UserId: The User Identifier (section 2.2.1.13) of the current user making the request for permissions.

@BelongsToGroup: A 1 indicates the User specified by **@UserId** belongs to the site group specified by **@GroupId**. This is used to determine access to group information, as groups can restrict membership information to members.

Copyright © 2014 Microsoft Corporation.

@CanViewMembership: A 1 indicates the user specified by **@UserId** is permitted to view site group membership.

@CanEditMembership: A 1 indicates the user specified by **@UserId** is permitted to edit site group membership.

@GroupOwnerId: Identifier of the user who owns the site group specified by **@GroupId**.

@IsExplicitlyInMembership: A 1 indicates the user specified by **@UserId** is a member of the site group specified by **@GroupId**, directly and not via an intermediate site group membership.

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

Value	Description
0	Successful execution.
1319	Group not found for the provided @GroupId and @SiteId.

The **proc_SecGetUserPermissionOnGroup** stored procedure MUST return zero result sets.

3.1.5.84 proc_SecListAllSiteMembers

The **proc_SecListAllSiteMembers** stored procedure provides information about all non-deleted security principals (2) in the specified site collection.

```
PROCEDURE proc_SecListAllSiteMembers (
      @SiteID                       uniqueidentifier
);
```

@SiteID: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the requested security principal (2) information.

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

The **proc_SecListAllSiteMembers** stored procedure MUST return one result set.

3.1.5.84.1 User Information Result Set

The User Information Result Set returns information about the non-deleted users directly associated with the specified site collection. Zero or more rows MUST be returned. There MUST be one row per non-deleted User in the site collection specified by **@**SiteId.

```
      tp_Id                         int,
      tp_SystemID                   varbinary(512),
      tp_Title                      nvarchar(255),
      tp_Login                      nvarchar(255),
      tp_Email                      nvarchar(255),
      tp_Notes                      nvarchar(1023),
      tp_SiteAdmin                  bit,
      tp_DomainGroup                bit;
```

tp_Id: The integer that uniquely identifies the security principal (2) with the site collection.

Copyright © 2014 Microsoft Corporation.

tp_SystemID: The **SystemID** (section 2.2.1.12) of the security principal (2).

tp_Title: The display name of the security principal (2).

tp_Login: The login name of the security principal (2).

tp_Email: The email address of the security principal (2), for example, "username@mail.example.com".

tp_Notes: Notes associated with the security principal (2).

tp_SiteAdmin: Set to 1 if the security principal (2) is a site administrator; otherwise, 0.

tp_DomainGroup: Set to 1 if the security principal (2) is a domain group; otherwise, 0.

3.1.5.85 proc_SecListAllWebMembers

The **proc_SecListAllWebMembers** stored procedure is invoked to list all non-deleted User and domain group accounts registered with a specified site (2).

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

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the specified site.

@WebId: The Site Identifier (section 2.2.1.11) of the site to query for members.

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

The **proc_SecListAllWebMembers** stored procedure MUST return a single result set:

3.1.5.85.1 Site Membership Result Set

The Site Membership Result Set returns a list of all non-deleted Users or domain groups registered with the specified Site. The Site Membership Result Set MUST return zero rows if the given Site has no members, otherwise it MUST contain one row for each member if **@**WebId matches an existing Site within the specified site collection.

```
      tp_Id                         int,
      tp_SystemID                   varbinary(512),
      tp_Title                      nvarchar(255),
      tp_Login                      nvarchar(255),
      tp_Email                      nvarchar(255),
      tp_Notes                      nvarchar(1023),
      tp_SiteAdmin                  bit,
      tp_DomainGroup                bit;
```

tp_ID: The User Identifier (section 2.2.1.13) of the user or domain group.

tp_SystemID: The **SystemID** (section 2.2.1.12) of the user or domain group.

tp_Title: The display name of the user or domain group.

Copyright © 2014 Microsoft Corporation.

tp_Login: The login name of the user or domain group.

tp_Email: The email address of the user or domain group.

tp_Notes: A descriptive text associated with the user or domain group.

tp_SiteAdmin: Contains a bit flag set to "1" if the user is a site administrator; otherwise, "0".

tp_DomainGroup: Contains a bit flag set to "1" if the User Identifier is for a domain group; otherwise, "0".

3.1.5.86 proc_SecListGroupsInRole

The **proc_SecListGroupsInRole** stored procedure is invoked to list all site groups that have the specified role.

```
PROCEDURE proc_SecListGroupsInRole(
      @SiteId                       uniqueidentifier,
      @ScopeId                      uniqueidentifier,
      @RoleId                       int
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection that contains the site groups that have the role.

@ScopeId: The Scope Identifier (section 2.2.1.8) of the scope containing the role.

@RoleId: Specifies the Role Identifier (section 2.2.1.7) for the role.

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

The **proc_SecListGroupsInRole** stored procedure MUST return one result set.

3.1.5.86.1 Site Group Information Result Set

The Site Group Information Result Set returns the collection of site groups that have the specified role. The Site Group Information Result Set MUST have one row per site group. The Site Group Information Result Set schema is defined in the Sec_SiteGroupsView (section 2.2.7.6).

3.1.5.87 proc_SecListScopeGroups

proc_SecListScopeGroups lists site groups which have role assignments in a specified scope.

```
PROCEDURE proc_SecListScopeGroups(
      @SiteId                       uniqueidentifier,
      @ScopeId                      uniqueidentifier
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection which contains the scope whose site groups with role assignments are being listed.

@ScopeId: The Scope Identifier (section 2.2.1.8) of the specified scope.

Copyright © 2014 Microsoft Corporation.

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

The **proc_SecListScopeGroups** stored procedure MUST return a single result set as defined in the following section.

3.1.5.87.1 Site Groups Result Set

The Site Groups Result Set contains all site group information for site groups with a role assignment in the specified scope. The Site Groups Result Set MUST be empty if either the site collection or scope are not found, or if no site groups have role assignments in the specified scope. Otherwise, one row MUST be returned for each site group in the site collection which has a role assignment in the specified scope.

The Site Groups Result Set is the same as the view defined in the Sec_SiteGroupsView (section 2.2.7.6).

3.1.5.88 proc_SecListScopeUsers

The **proc_SecListScopeUsers** stored procedure is invoked to list (1) information about users and domain groups with a direct role assignment association with the specified scope, rather than by membership in a site group with a role assignment association with the scope.

```
PROCEDURE proc_SecListScopeUsers(
      @SiteId                       uniqueidentifier,
      @ScopeId                      uniqueidentifier
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the scope.

@ScopeId: The Scope Identifier (section 2.2.1.8) of the scope for which information about users with role assignments is requested.

Return Values: The **proc_SecListScopeUsers** stored procedure MUST return an integer return code of 0, and it MUST return a single result set with zero or more rows.

3.1.5.88.1 User Information Result Set

The User Information Result Set returns information about the Users and domain groups directly associated with the specified scope through a role assignment. The User Information Result Set MUST contain one row for each non-deleted User or domain group directly associated with the specified scope through a role assignment. The User Information Result Set MUST contain zero rows if no Users or domain groups are directly associated with the specified scope, or if the parameters @SiteId or @ScopeId do not match existing site collection or Scope Identifiers (section 2.2.1.8), respectively.

The User Information Result Set is defined in the Common Result Sets Principal User Information Result Set (section 2.2.5.17).

3.1.5.89 proc_SecListSiteGroupMembership

The **proc_SecListSiteGroupMembership** stored procedure lists members in a specified site group. To view membership, the site group MUST be set to allow everyone to view membership, or the current user MUST belong to the site group, be a site auditor, or be the Site owner.

Copyright © 2014 Microsoft Corporation.

```
PROCEDURE proc_SecListSiteGroupMembership (
      @SiteId                       uniqueidentifier,
      @GroupId                      int,
      @CurrentUserId                int,
      @SiteAuditor                  bit,
      @BelongsToGroup               bit,
      @GroupOwnerId                 int,
      @CurrentUserIsOwner           bit
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the site group.

@ GroupId: The identifier of the site group.

@ CurrentUserId: The identifier of the current user. This is used to determine privileges.

@ SiteAuditor: A bit set to 1 if the current user is a site auditor. This is used to determine privileges.

@ BelongsToGroup: A bit set to 1 if the current user is a member of the site group. This is used to determine privileges.

@ GroupOwnerId: The identifier of the owner of the site group. This is used to determine privileges.

@ CurrentUserIsOwner: A bit set to 1 if the current user is the owner of the site group. This is used to determine privileges.

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

Value	Description
0	Successful execution.
5	The User does not have privileges to view site group membership.

The **proc_SecListSiteGroupMembership** stored procedure MUST return a site group Membership result set.

3.1.5.89.1 User Information Result Set

The User Information Result Set MUST either be empty or include one row of information for each member of the site group. If the User specified by @CurrentUserId does not have privileges to view membership, then the User Information Result Set will be empty. To view membership, the current user MUST belong to the site group, be a Site auditor, or be the Site owner.

The User Information Result Set is defined the Common Result Sets Principal User Information Result Set (section 2.2.5.17).

3.1.5.90 proc_SecListSiteGroups

The **proc_SecListSiteGroups** stored procedure is invoked to list site group information for a specified site collection.

Copyright © 2014 Microsoft Corporation.

```
PROCEDURE proc_SecListSiteGroups(
      @SiteId                       uniqueidentifier,
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the site groups to list.

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

The **proc_SecListSiteGroups** stored procedure MUST return one result set as follows.

3.1.5.90.1 Site Group Information Result Set

The Site Group Information Result Set contains all site group information for the site collection. The Site Group Information Result Set MUST be empty if the site collection is not found. Otherwise, one row MUST be returned for each site group in the specified site collection. The Site Group Information Result Set is defined using T-SQL syntax, in the Sec_SiteGroupsView (section 2.2.7.6) section.

3.1.5.91 proc_SecListSiteGroupsContainingUser

The **proc_SecListSiteGroupsContainingUser** stored procedure lists the site groups in which the user is a member.

```
PROCEDURE proc_SecListSiteGroupsContainingUser(
      @SiteId                       uniqueidentifier,
      @UserId                       int
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the site groups to be listed. This parameter MUST correspond to a valid site collection, and MUST NOT be NULL.

@UserId: The User Identifier (section 2.2.1.13) used to find site groups that contain this User.

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

The **proc_SecListSiteGroupsContainingUser** stored procedure MUST return one result set.

3.1.5.91.1 Site Group Information Result Set

The Site Group Information Result Set contains information of site groups where the User is a member. The Site Group Information Result Set MUST be returned, and MUST contain one row for each site group found for the User. The Site Group Information Result Set schema is defined in the Sec_SiteGroupsView (section 2.2.7.6).

3.1.5.92 proc_SecListSiteGroupsWhichUserOwns

The **proc_SecListSiteGroupsWhichUserOwns** stored procedure returns site group information for the site groups owned by the specified principal (1).

```
PROCEDURE proc_SecListSiteGroupsWhichUserOwns(
      @SiteId                       uniqueidentifier,
```


Copyright © 2014 Microsoft Corporation.

```
      @UserId                       int
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the principal (1) and the site groups to be listed.

@UserId: The User Identifier (section 2.2.1.13) of the principal (1), used to find the site groups it owns.

Return Code Values: The **proc_SecListSiteGroupsWhichUserOwns** stored procedure returns an integer which MUST be 0.

The **proc_SecListSiteGroupsWhichUserOwns** stored procedure MUST return the following result set:

3.1.5.92.1 Site Group Information Result Set

The Site Group Information Result Set returns site group information for all site groups which are owned either directly by the User or external group specified by **@**UserId, or for site groups which are owned indirectly, when the User or external group is a member of a site group which is the returned site group's owner.

The Site Group Information Result Set MUST contain one row for each site group owned. The T-SQL syntax for the Site Group Information Result Set is defined in the Sec_SiteGroupsView (section 2.2.7.6).

3.1.5.93 proc_SecListUsersInRole

The **proc_SecListUsersInRole** stored procedure lists the non-deleted principals assigned to a specified role in the specified scope.

```
PROCEDURE proc_SecListUsersInRole(
      @SiteId                       uniqueidentifier,
      @ScopeId                      uniqueidentifier,
      @RoleId                       int
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the scope, role, and principals to be listed.

@ScopeId: The Scope Identifier (section 2.2.1.8) of the scope within which to match principals assigned to the role.

@RoleId: Specifies the Role Identifier (section 2.2.1.7) of the role definition assigned to the principals to be listed.

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

The **proc_SecListUsersInRole** stored procedure MUST return one result set.

Copyright © 2014 Microsoft Corporation.

3.1.5.93.1 User Role Membership Result Set

The User Role Membership Result Set returns information about the non-deleted principals assigned to the specified role in the specified scope and site collection. The User Role Membership Result Set MUST be empty if no principals assigned to the role are found; otherwise it MUST include one row of information for each matching principal (1).

```
tp_Id                       int,
tp_SystemID                 varbinary(512),
tp_Title                    nvarchar(255),
tp_Login                    nvarchar(255),
tp_Email                    nvarchar(255),
tp_Notes                    nvarchar(1023),
tp_SiteAdmin                bit,
tp_DomainGroup              bit;
```

tp_Id: The User Identifier (section 2.2.1.13) of the principal (1).

tp_SystemID: The **SystemID** (section 2.2.1.12) of the principal (1).

tp_Title: The display name of the principal (1).

tp_Login: The login name of the principal (1).

tp_Email: The email address of the principal (1).

tp_Notes: Contains notes about the principal (1).

tp_SiteAdmin: 1 if the principal (1) is a site administrator, otherwise 0.

tp_DomainGroup: 1 if the principal (1) is a domain group, otherwise 0.

3.1.5.94 proc_SecMigrateUser

The **proc_SecMigrateUser** stored procedure updates the **SystemID** (section 2.2.1.12) and login name for a specified user (that is, a principal (1) that is not a domain group) in the UserInfo Table. If an existing User is found with the new login name and **SystemID**, **proc_SecMigrateUser** MUST update that User with a new unique **SystemID** and mark that User as deleted in the UserInfo table before updating the specified User.

```
PROCEDURE proc_SecMigrateUser(
      @SiteId                   uniqueidentifier,
      @OldLogin                 nvarchar(255),
      @NewLogin                 nvarchar(255),
      @NewSystemId              varbinary(512),
      @IsWindowsAuth            bit,
      @OldSystemId              varbinary(512)      = NULL
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the User to be updated.

@OldLogin: The current value of the user's login name.

@NewLogin: The new value to set as the user's login name. This parameter MUST NOT be NULL.

Copyright © 2014 Microsoft Corporation.

@NewSystemId: The new **SystemID** to set for the User. This parameter MUST NOT be NULL.

@IsWindowsAuth: A bit specifying whether the User is authenticated with Microsoft Windows-integrated authentication. If this parameter is set to 1 or NULL, the User is authenticated using Windows-integrated authentication.

@OldSystemId: The current value of the user's **SystemID**. This parameter can be NULL. If the **@OldSystemId** parameter is not NULL, then the User specified by **@OldLogin** MUST also have the **SystemID** specified by **@OldSystemId**.

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

Value	Description
0	Successful execution.
1003	An existing User was found with the new login name and **SystemID**, but proc_SecMigrateUser failed to update it with a new **SystemID** and mark it as deleted. Failed to find or update the specified User. The principal (1) found with @NewLogin is a domain group. @IsWindowsAuth is non-zero (not using Windows Authorization)

The **proc_SecMigrateUser** stored procedure MUST return one result set if execution is successful; otherwise, zero result sets are returned.

3.1.5.94.1 User Migration Info Result Set

The User Migration Info Result Set returns User Identifiers (section 2.2.1.13) for an existing User with the new login name and **SystemID** (section 2.2.1.12) that has been marked deleted, if any, and for the specified User that has been updated with the new login name and **SystemID**. The User Migration Info Result Set MUST be returned if execution is successful; otherwise, the User Migration Info Result Set MUST NOT be returned. When the User Migration Info Result Set is returned, it MUST contain one row.

```
        {OldUserId}                     int,
        {NewUserId}                     int;
```

{OldUserId}: Contains the User Identifier of the specified User whose login name and **SystemID** have been updated with the new values.

{NewUserId}: SHOULD contain -1 to indicate that an existing User with the new login name and **SystemID** was not found.

3.1.5.95 proc_SecReCalculateWebFGP

The **proc_SecReCalculateWebFGP** stored procedure is invoked to update the bit at 0x00000400 of the Site Property Flags (section 2.2.2.10) to indicate whether the site (2) has at least one uniquely secured object within it. This bit can be used by a search tool to determine whether it SHOULD avoid indexing pages in this Site so as not to reveal secure information.

```
  PROCEDURE proc_SecReCalculateWebFGP(
        @SiteId                         uniqueidentifier,
        @WebId                          uniqueidentifier
```


Copyright © 2014 Microsoft Corporation.

```
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the Site to be updated. This parameter MUST correspond to a valid site collection, and MUST NOT be NULL.

@WebId: The Site Identifier (section 2.2.1.11) of the site (2) containing the flag to be updated.

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

The **proc_SecReCalculateWebFGP** stored procedure MUST return zero result sets.

3.1.5.96 proc_SecRefreshToken

The **proc_SecRefreshToken** stored procedure is invoked to update the UserInfo Table with information about a specified user's membership in external groups.

```
PROCEDURE proc_SecRefreshToken(
      @SiteId                       uniqueidentifier,
      @UserId                       int,
      @ExternalToken                image,
      @ExternalTokenTime            datetime
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the User to update in the UserInfo Table.

@UserId: The User Identifier (section 2.2.1.13) of the User.

@ExternalToken: An External Group Token (section 2.2.4.2) containing external group membership information for the User. This value can be NULL, specifying that the User is not known to be a member of any external groups. The **proc_SecRefreshToken** SHOULD update the UserInfo Table row, matching the User Identifier with this value in the **tp_ExternalToken** field (2).

@ExternalTokenTime: A datetime in UTC specifying when the External Group Token in the **@ExternalToken** parameter was last updated. This parameter MUST be NULL if the @ExternalToken parameter is NULL. The **proc_SecRefreshToken** SHOULD update the UserInfo Table row, matching the User Identifier with this value in the **tp_ExternalTokenLastUpdated** field (2).

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

The **proc_SecRefreshToken** stored procedure MUST NOT return a result set.

3.1.5.97 proc_SecRemoveGroup

The **proc_SecRemoveGroup stored** procedure is invoked to remove a site group from a site collection.

```
PROCEDURE proc_SecRemoveGroup(
      @SiteId                       uniqueidentifier,
      @GroupId                      int,
      @UserID                       int,
      @SiteAdmin                    bit,
```


Copyright © 2014 Microsoft Corporation.

```
        @GroupOwnerId                   int,
        @CurrentUserIsOwner             bit
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the site group to remove.

@GroupId: The identifier for the site group to be removed.

@UserID: The User Identifier (section 2.2.1.13) for the current user. Proc_SecRemoveGroup MUST re-assign ownership of any site group previously owned by the deleted site group to this User Identifier.

@SiteAdmin: A bit flag specifying whether the current user is a site administrator.

- If this flag is set to 1, proc_SecRemoveGroup MUST remove the site group.
- If this flag is set to 0, proc_SecRemoveGroup MUST NOT remove the site group unless another parameter specifies that it MUST.

@GroupOwnerId: The User Identifier or Site Group Identifier (section 2.2.1.10) of the site group owner.

- If **@CurrentUserIsOwner** is set to 1 this value MUST be set to the User Identifier of the current user, or a site group which includes the current user.
- If **@CurrentUserIsOwner** is set to 0, this value MUST be ignored.

@CurrentUserIsOwner: A bit flag specifying that the current user is an owner of the site group.

- If **@CurrentUserIsOwner** is set to 1 and **@GroupOwnerId** contains the User Identifier of the owner of the site group, or the Site Group Identifier of a site group which includes the current user, proc_SecRemoveGroup MUST remove the site group.
- If **@CurrentUserIsOwner** is set to 0, proc_SecRemoveGroup MUST NOT remove the site group unless another parameter specifies that it MUST.

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

Value	Description
0	Successful execution.
5	The current user has insufficient permissions to remove the site group.

The **proc_SecRemoveGroup stored** procedure returns either zero or one result set. If the site group is successfully removed, proc_SecRemoveGroup MUST return a DLAlias Result Set. If the current user does not have sufficient permissions to remove the site group, **proc_SecRemoveGroup** MUST NOT return a result set.

3.1.5.97.1 DLAlias Result Set

If the site group existed and was removed successfully, the DLAlias Result Set MUST return the email distribution address of the site group in a single row. The DLAlias Result Set MUST be empty if the email distribution address was blank.

Copyright © 2014 Microsoft Corporation.

```
      DLAlias                         varchar(128);
```

DLAlias: The email distribution address for the site group.

3.1.5.98 proc_SecRemovePrincipalFromScope

The **proc_SecRemovePrincipalFromScope** stored procedure removes a principal (1) from a security role associated with a site (2) in a specified scope. If no role is specified, or if the role specified is the "Guest" Role, the principal (1) MUST be removed from all roles in the scope, and it MUST also be removed from all roles in other specified scopes or subsite contained within the Site.

```
PROCEDURE proc_SecRemovePrincipalFromScope (
      @SiteId                         uniqueidentifier,
      @WebId                          uniqueidentifier,
      @WebScopeId                     uniqueidentifier,
      @ScopeId                        uniqueidentifier,
      @PrincipalId                    int,
      @RoleId                         int = NULL
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the principal (1) and scope.

@WebId: The Site Identifier (section 2.2.1.11) of the site (2) containing the principal (1) and scope.

@WebScopeId: The Scope Identifier (section 2.2.1.8) of the scope which includes the Site. If the Site has unique permissions, this MUST be the scope of the Site. If **@RoleId** is not NULL and not the Role Identifier (section 2.2.1.7) of the Guest Role(1073741825), then **@WebScopeId** is ignored. Otherwise, **@WebScopeId** determines additional scopes or Sites from which the principal (1) MUST be removed, as follows.

Value	Meaning
@WebScopeId = @ScopeId	Remove the principal (1) from the specified role or roles within the Site and within all of the Site's subsites which share the same scope.
@WebScopeId ≠ @ScopeId	Remove the principal (1) from the specified role or roles in all scopes contained within the Site.

@ScopeId: The Scope Identifier of the scope in which to remove the principal (1) from the role. See **@**WebScopeId for behavior based on comparison of the **@ScopeId** and **@WebScopeId** parameter values.

@PrincipalId: The User Identifier (section 2.2.1.13) of the principal (1) to remove.

@RoleId: The Role Identifier of the role to remove the principal (1) from, within the specified scopes. The value in **@RoleId** MUST either correspond to a valid role or be NULL. If this value is NULL, or if **@RoleID** is the Role Identifier of the Guest Role (1073741825), **proc_SecRemovePrincipalFromScope** MUST remove the principal (1) from all roles in the specified scopes.

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

Copyright © 2014 Microsoft Corporation.

The **proc_SecRemovePrincipalFromScope** stored procedure MUST return a single result set as follows.

3.1.5.98.1 Site Audit Mask Result Set

The Site Audit Mask Result Set contains the information about the Audit Flags (section 2.2.2.1) associated with the specified Site. The Site Audit Mask Result Set MUST be returned and MUST contain a single row. The Site Audit Mask Result Set is defined in the Common Result Sets Site Audit Mask Result Set (section 2.2.5.20).

3.1.5.99 proc_SecRemoveRoleDef

The **proc_SecRemoveRoleDef** stored procedure removes a role definition, and any role assignments which use that role definition, from a specified site (2).

```
PROCEDURE proc_SecRemoveRoleDef(
      @SiteId                       uniqueidentifier,
      @WebId                        uniqueidentifier,
      @RoleId                       int
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the role definition to be removed.

@WebId: The Site Identifier (section 2.2.1.11) of the Site associated with the role definition to be removed.

@RoleId: The identifier of the role definition to be removed.

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

The **proc_SecRemoveRoleDef** stored procedure MUST return the following result set.

3.1.5.99.1 Site Audit Mask Result Set

The Site Audit Mask Result Set contains information about the Audit Flags (section 2.2.2.1) associated with the Site. The Site Audit Mask Result Set MUST be returned and MUST contain one row. The Site Audit Mask Result Set is defined in the Common Result Sets Site Audit Mask Result Set (section 2.2.5.20).

3.1.5.100 proc_SecRemoveUserFromScopeByLogin

The **proc_SecRemoveUserFromScopeByLogin** stored procedure is invoked to remove a user, specified by login name, from a role in a specified scope. If no role is specified, or if the role specified is the "Guest" Role, then the user will be removed from all roles in the specified scope and will also be removed from other scopes associated with the specified site (2).

```
PROCEDURE proc_SecRemoveUserFromScopeByLogin(
      @SiteId                       uniqueidentifier,
      @WebId                        uniqueidentifier,
      @WebScopeId                   uniqueidentifier,
      @ScopeId                      uniqueidentifier,
      @LoginName                    nvarchar(255),
      @RoleId                       int               = NULL
```


Copyright © 2014 Microsoft Corporation.

```
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the User and scope.

@WebId: The Site Identifier (section 2.2.1.11) of the site (2) using the role.

@WebScopeId**:** The Scope Identifier (section 2.2.1.8) of a scope in the site collection. If **@RoleId** is not NULL and is not the Role Identifier (section 2.2.1.7) of the "Guest" Role (1073741825), then **@WebScopeId** is ignored. Otherwise, **@WebScopeId** determines which additional scopes the User MUST be removed from, as follows:

- If **@WebScopeId** is equal to **@ScopeId**, remove the specified User from all scopes associated with all Sites sharing the permissions of the Site specified by **@**WebId.
- If **@WebScopeId** is not equal to **@ScopeId**, remove the specified User from all scopes associated with the Site specified by **@WebId**.

@ScopeId: The Scope Identifier of a scope in the site collection. See **@WebScopeId** for conditions of equality between **@ScopeId** and **@WebScopeId**.

@LoginName: The login name of the User.

@RoleId: The Role Identifier to remove from the specified scopes. **@RoleId** MUST be NULL or correspond to a valid role. If **@RoleId** is NULL, or is the Role Identifier of the Guest Role (1073741825), **proc_SecRemoveUserFromScopeByLogin** MUST remove the User corresponding to **@**LoginName from all roles.

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

Value	Description
0	Successful execution.
1317	UserId not found for site collection or login name.

The **proc_SecRemoveUserFromScopeByLogin** stored procedure MUST return a single result set as follows.

3.1.5.100.1 Site Audit Mask Result Set

The Site Audit Mask Result Set returns audit information about the Site specified by **@**WebId. The Site Audit Mask Result Set MUST be returned, and MUST contain only one row, as defined in the Common Result Sets Site Audit Mask Result Set (section 2.2.5.20).

3.1.5.101 proc_SecRemoveUserFromSite

The **proc_SecRemoveUserFromSite** stored procedure is invoked to remove a user from a site collection. If the User owns groups, **proc_SecRemoveUserFromSite** will assign group ownership to the specified new group owner.

```
PROCEDURE proc_SecRemoveUserFromSite(
      @SiteId                       uniqueidentifier,
      @NewSiteGroupOwnerId          int,
```


Copyright © 2014 Microsoft Corporation.

```
      @UserId                         int
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) for the site collection containing the User to be removed.

@NewSiteGroupOwnerId: An identifier for the User who will take over group ownership from the User being removed. If **@NewSiteGroupOwnerId** has the same value as **@UserId**, the ownership change will not succeed. To succeed, this value MUST be -1 or a valid User Identifier (section 2.2.1.13) in the specified site collection. If the value is -1, the site collection owner will be the new site group owner.

@UserId: The User Identifier for the User who is being removed. If **@UserId** is the site collection owner or secondary contact, the change will not succeed. To succeed, this value MUST be an existing User Identifier for the specified site collection.

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

Value	Description
0	0 is returned whenever the conditions for the alternate return values of 16 and 4335 are not met.
16	@NewSiteGroupOwnerId equals @UserId. No changes were made.
4335	@UserId is the site collection owner or secondary contact. No changes were made.

The **proc_SecRemoveUserFromSite** stored procedure MUST NOT return any result sets.

3.1.5.102 proc_SecRemoveUserFromSiteGroup

The **proc_SecRemoveUserFromSiteGroup** stored procedure is invoked to remove a user from a site group in a site collection.

```
PROCEDURE proc_SecRemoveUserFromSiteGroup(
      @SiteId                         uniqueidentifier,
      @GroupId                        int,
      @UserIDToBeDeleted              int,
      @UserID                         int,
      @SiteAdmin                      bit,
      @BelongsToGroup                 bit,
      @GroupOwnerId                   int,
      @CurrentUserIsOwner             bit
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection that will be queried for the requested Group.

@GroupId: The Site Group Identifier (section 2.2.1.10) for the site group that the User specified by **@UserIDToBeDeleted** is to be removed from.

@UserIDToBeDeleted: The User Identifier (section 2.2.1.13) for the User to remove from the specified site group. If **@UserId** and **@UserIdToBeDeleted** refer to the same User, and the site group permits users to remove themselves from site group, membership, the User specified by **@UserIdToBeDeleted** MUST be removed from the group.

Copyright © 2014 Microsoft Corporation.

@UserID: The User Identifier for the current user who is removing the User to be deleted from the specified site group. If **@UserId** and **@UserIdToBeDeleted** refer to the same User, and the site group Site Group permits site group members to edit membership, the User specified by **@UserIdToBeDeleted** is removed from the group.

@SiteAdmin: A bit flag specifying whether the User specified by **@UserID** is a site administrator of the site (2) specified by **@SiteId**. If **@SiteAdmin** is set to "1", **proc_SecRemoveUserFromSiteGroup** MUST remove the User specified by **@UserIDToBeDeleted** from the site group.

@BelongsToGroup: A bit flag specifying whether the User specified by **@UserID** is a member of the site group specified by **@GroupId**. If **@BelongsToGroup** is set to "1" and the site group permits site group members to edit membership, and the current user specified in **@UserID** is a member of the site group, **proc_SecRemoveUserFromSiteGroup** MUST remove the User specified by **@UserIDToBeDeleted** from the site group.

@GroupOwnerId: The User Identifier or Site Group Identifier of the site group owner specified by **@GroupId**. **@GroupOwnerId** MUST be set to the User Identifier of the current user, or, if @CurrentUserIsOwner is set to "1," to the Site Group Identifier of a site group which includes the current user. If **@CurrentUserIsOwner** is set to "0", **@GroupOwnerId** MUST be ignored.

@CurrentUserIsOwner: A bit flag specifying that the User specified by **@GroupOwnerId** is an owner of the site group specified by **@GroupId**. If **@CurrentUserIsOwner** is set to "1" and **@GroupOwnerId** contains either the User Identifier of the owner of the site group, or the Site Group Identifier of a site group which includes the current user, **proc_SecRemoveUserFromSiteGroup** MUST remove the User from the site group.

Return Values: The **proc_SecRemoveUserFromSiteGroup** stored procedure returns an integer return code which MUST be one of the values in the following table.

Value	Description
0	Successful execution.
5	The current user has insufficient authority to remove the User from the site group.

The **proc_SecRemoveUserFromSiteGroup** stored procedure MUST NOT return any result sets.

3.1.5.103 proc_SecRemoveUserFromSiteGroupByLogin

The **proc_SecRemoveUserFromSiteGroupByLogin** stored procedure is invoked to remove a user identified by login name from a specified site group.

```
PROCEDURE proc_SecRemoveUserFromSiteGroupByLogin(
      @SiteId                       uniqueidentifier,
      @GroupId                      int,
      @LoginName                    nvarchar(255),
      @UserID                       int,
      @SiteAdmin                    bit,
      @BelongsToGroup               bit,
      @GroupOwnerId                 int,
      @CurrentUserIsOwner           bit,
);
```


Copyright © 2014 Microsoft Corporation.

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the site group and the User to be removed.

@GroupId: The identifier of the site group containing the User to be removed.

@LoginName: The login name of the User to be removed from the site group.

@UserID: The User Identifier (section 2.2.1.13) of the current user.

@SiteAdmin: A bit flag specifying whether the User specified by **@UserID** is a site administrator of the site (2) specified by **@SiteId**. If **@SiteAdmin** is set to "1", and **@LoginName** specifies an existing user, **proc_SecRemoveUserFromSiteGroupByLogin** MUST remove the User specified by **@LoginName** from the site group.

@BelongsToGroup: A bit flag specifying whether the User specified by **@UserID** is a member of the site group specified by **@GroupId**. If **@BelongsToGroup** is set to "1" and the site group permits users to remove themselves from site group membership, and the current user specified in **@UserID** is a member of the site group, and **@LoginName** specifies an existing user, **proc_SecRemoveUserFromSiteGroupByLogin** MUST remove the User specified by **@LoginName** from the site group.

@GroupOwnerId: The User Identifier or Site Group Identifier (section 2.2.1.10) of the site group owner specified by **@GroupId**. **@GroupOwnerId** MUST be set to the User Identifier of the current user, or, if @CurrentUserIsOwner is set to "1," to the Site Group Identifier of a site group which includes the current user. If **@CurrentUserIsOwner** is set to "0", **@GroupOwnerId** MUST be ignored.

@CurrentUserIsOwner: A bit flag specifying that the User specified by **@GroupOwnerId** is an owner of the site group specified by **@GroupId**. If **@CurrentUserIsOwner** is set to "1" and **@GroupOwnerId** contains either the User Identifier of the owner of the site group, or the Site Group Identifier of a site group which includes the current user, and **@LoginName** specifies an existing user, **proc_SecRemoveUserFromSiteGroupByLogin** MUST remove the User specified by **@LoginName** from the site group.

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

Value	Description
0	Successful execution.
5	The current user has insufficient permissions to remove the specified User from the site group.
1317	UserId not found for site collection or login name.

The **proc_SecRemoveUserFromSiteGroupByLogin** stored procedure MUST NOT return a result set.

3.1.5.104 proc_SecResetItemPerm

The **proc_SecResetItemPerm** stored procedure is invoked to cause a list item to revert its set of permissions so that it inherits permissions from its parent rather than has its own. The previous set of unique permissions for the list item are discarded.

```
PROCEDURE proc_SecResetItemPerm(
      @SiteId                         uniqueidentifier,
```


Copyright © 2014 Microsoft Corporation.

```
    @WebId                          uniqueidentifier,
    @OldScopeId                     uniqueidentifier,
    @Url                            nvarchar(260),
    @DocId                          nvarchar(260),
    @NewScopeId                     uniqueidentifier = NULL OUTPUT
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the list item.

@WebId: The Site Identifier (section 2.2.1.11) of the site containing the item.

@OldScopeId: The scope identifier of the scope of the item.

@Url: The store relative-form URL of the item.

@DocId: The document identifier (section 2.2.1.2) of the document.

@NewScopeId: This is an output parameter, which provides the Scope Identifier (section 2.2.1.8) of the scope of the item. This MUST be the Scope Identifier of the item's parent container upon successful execution; otherwise, it remains unchanged.

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

Value	Description
0	Successful execution.
3	The item specified is not configured to have unique permissions. This value MUST also be returned when the list item is not found at the specified location, or the @OldScopeID does not match the specified list item's scope.

Upon successful execution, the **proc_SecResetItemPerm** stored procedure MUST return one result set as follows.

3.1.5.104.1 Site Audit Mask Result Set

The Site Audit Mask Result Set contains the information about the Audit Flags (section 2.2.2.1) associated with the specified Site. It MUST contain a single row. The Site Audit Mask Result Set is defined using T-SQL syntax, as defined in the Common Result Sets Site Audit Mask Result Set (section 2.2.5.20).

3.1.5.105 proc_SecResetWebToDefaultRoleDefinition

The **proc_SecResetWebToDefaultRoleDefinition** stored procedure is invoked to replace existing roles, role assignments and Permissions for a specified site (2) and its subsites with a default set of role definitions, as specified by input parameters. If the specified Site inherits its security permissions, a new scope will be made to hold the new security settings. **proc_SecResetWebToDefaultRoleDefinition** will fail if the specified Site does not exist or if the specified Site has a unique role definition.

```
PROCEDURE proc_SecResetWebToDefaultRoleDefinition(
    @WebId                          uniqueidentifier,
    @AuthorID                       int,
```


Copyright © 2014 Microsoft Corporation.

```
        @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,
        @AnonymousPermMask              bigint,
        @ScopeId                        uniqueidentifier OUTPUT
);
```

@WebId: The Site Identifier (section 2.2.1.11) of the Site.

@AuthorID: The user identifier of the principal (1) to be assigned to the Administrator Role. This MUST be the user identifier of an existing principal (1) in the site collection.

@AdminsName: The display name for the Administrator Role. This value MUST NOT be NULL.

@AdminsDescription: The description for the Administrator Role.

@AdminsPermMask: The permissions mask for the Administrator Role. This value MUST NOT be NULL.

@AuthorsName: The display name for the Web Designer Role. This value MUST NOT be NULL.

@AuthorsDescription: The description for the Web Designer Role.

@AuthorsPermMask: The permissions mask for the Web Designer Role. This value MUST NOT be NULL.

@ContributorsName: The display name for the Contributor Role. This value MUST NOT be NULL.

@ContributorsDescription: The description for the Contributor Role.

@ContributorsPermMask: The permissions mask for the Contributor Role. This value MUST NOT be NULL.

@BrowsersName: The display name for the Reader Role. This value MUST NOT be NULL.

@BrowsersDescription: The description for the Reader Role.

@BrowsersPermMask: The permissions mask for the Reader Role. This value MUST NOT be NULL.

@GuestsName: The display name for the Guest Role. This value MUST NOT be NULL.

@GuestsDescription: The description for the Guest Role.

@GuestsPermMask: The permissions mask for the Guest Role. This value MUST NOT be NULL.

Copyright © 2014 Microsoft Corporation.

@AnonymousPermMask: A WSS Rights Mask (section 2.2.2.13) which indicates the rights granted to a User that is anonymous, or has no specific rights.

@ScopeId: The scope identifier of the scope of the updated permissions, returned as an output parameter.

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

Value	Description
0	Successful execution.
3	Site does not exist or Site has a unique role definition associated with it.

If execution is not successful, the **proc_SecResetWebToDefaultRoleDefinition** stored procedure MUST return no result sets. Otherwise, **proc_SecResetWebToDefaultRoleDefinition** MUST return the Site Audit Mask Result Set, repeated six times.

3.1.5.105.1 Site Audit Mask Result Set

The Site Audit Mask Result Set is returned six times upon successful execution. The first is returned after the Administrator Role has been added, the second after the principal (1) specified by @AuthorId has been newly assigned to the administrator role and scope, and then once per additional role that is added.

Each Site Audit Mask Result Set contains the Audit Flags (section 2.2.2.1) of the Site and MUST hold a single row, and is defined in the Common Result Sets Site Audit Mask Result Set (section 2.2.5.20).

3.1.5.106 proc_SecResolvePrincipal

The **proc_SecResolvePrincipal** stored procedure retrieves information about a principal (1) or site group in the specified site collection, given the principal's login name, email address, or display name.

```
PROCEDURE proc_SecResolvePrincipal(
      @SiteId                       uniqueidentifier,
      @Input                        nvarchar(255),
      @InputIsEmailOnly             bit,
      @AccountName                  nvarchar(255),
      @DLAlias                      nvarchar(128),
      @DLAliasServerAddress         nvarchar(255),
      @SearchScope                  int
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection.

@Input: The principal's display name, email address, or login name. This parameter is used to find the principal (1).

@InputIsEmailOnly: Specifies how input parameters are used to find the principal (1). If this parameter is 0, then matching in the following order:

1. **@Input** to display name;

Copyright © 2014 Microsoft Corporation.

2. **@Input** as login name;

3. **@DLAlias** as email address.

For any value that is not 0, skip steps 1 and 2.

@AccountName: The principal's login name. If NULL, and **@Input** did not match on display name or login name, the value of **@**Input will be taken as login name.

@DLAlias: The principal's email alias to search for.

@DLAliasServerAddress: The server portion of an email address. If the principal (1) is a site group whose email address is available, this is used to construct the full email address in the {Email} column in the Principal Information Result Set (section 3.1.5.106.1). If this parameter is NULL and the principal (1) is a site group, {Email} MUST return NULL.

@SearchScope: A bit mask signifying the scope in which to search for matching principals. This can have one or more flags set. The following bits within **@SearchScope** can be set to 1.

Value	Description
0x01	user.
0x04	NT domain group.
0x08	Site group.

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

If a match for the principal (1) is found, the **proc_SecResolvePrincipal** stored procedure MUST return the Principal Information Result Set. Otherwise, it MUST NOT return a result set.

3.1.5.106.1 Principal Information Result Set

The Principal Information Result Set MUST return a single row containing information about the principal (1) which uniquely matches the input parameters, or special values described below which indicate that no match was found.

```
{MatchType}                    int,
{PrincipalType}                int,
{Id}                           int,
{Login}                        nvarchar(255),
{Email}                        nvarchar(255),
{DisplayName}                  nvarchar(255);
```

{MatchType}: Designates the type of match found between input variables and a principal (1) in the site group. Values MUST be listed in the following table.

Value	Description
1	Matched by login name.
2	Matched by email address.
4	Matched by display name.

Copyright © 2014 Microsoft Corporation.

{PrincipalType}: Designates the type of the principal (1). Value MUST be listed in the following table.

Value	Description
1	Domain User.
4	Domain Group.
8	Site group.

{Id}**:** The identifier for the principal. If no match was found then the value is -1.

{Login}: The login name of the principal (1). This parameter is NULL if no match was found.

{Email}: The email address, for example, name**@**mail.constco.com, of the principal (1). This parameter is NULL if no match was found, if no email address is available, or if the principal is a site group but **@DLAliasServerAddress** was NULL.

{DisplayName}: The display name of the principal (1). The value is NULL if no match was found.

3.1.5.107 proc_SecSetSiteGroupProperties

The **proc_SecSetSiteGroupProperties** stored procedure is invoked to update properties of a site group.

```
PROCEDURE proc_SecSetSiteGroupProperties(
      @SiteId                       uniqueidentifier,
      @GroupId                      int,
      @Title                        nvarchar(255),
      @Description                  nvarchar(512),
      @OwnerID                      int,
      @OwnerIsUser                  bit,
      @UserID                       int,
      @SiteAdmin                    bit,
      @GroupOwnerId                 int,
      @CurrentUserIsOwner           bit,
      @DLAlias                      nvarchar(255),
      @DLErrorMessage               nvarchar(255),
      @DLFlags                      int,
      @DLJobId                      int,
      @DLArchives                   varchar(4000),
      @RequestEmail                 nvarchar(255),
      @Flags                        int
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the site group to update.

@GroupId: The identifier for the site group to update.

@Title: The new user-friendly display name for the site group. If this parameter is NULL, the update MUST fail and the return code MUST be 80. However, if the user does not have sufficient permissions, the return code MUST be 5.

@Description: The new description for the site group. This parameter can be NULL.

Copyright © 2014 Microsoft Corporation.

@OwnerID: The identifier of a user, domain group, or site group to be set as the owner of the site group. This parameter MUST contain the identifier of an existing User, domain group, or site group.

@OwnerIsUser: A bit which specifies whether the value in **@OwnerID** is a User or domain group rather than a site group. A value of "1" specifies that the value in **@OwnerID** is a User or domain group. A value of "0" specifies that the value in **@OwnerID** is a site group.

@UserID: The User Identifier (section 2.2.1.13) of the current user. This parameter can be ignored by **proc_SecSetSiteGroupProperties**.

@SiteAdmin: A bit flag specifying whether the current user has site collection administration permissions. If this parameter is set to "1" to specify the current user has site collection administration permissions, site group management permissions MUST be granted. This parameter can be NULL.

@GroupOwnerId: The User Identifier or Site Group Identifier (section 2.2.1.10) to compare with the existing owner of the site group. Site group management permissions MUST be granted if the value in **@GroupOwnerId** matches the identifier of the existing owner of the site group and the value of **@CurrentUserIsOwner** is "1". This parameter can be NULL.

@CurrentUserIsOwner: A bit flag specifying whether the value in **@GroupOwnerId** is to be compared with the owner of the site group. Site group management permissions MUST be granted if the value in **@GroupOwnerId** matches the identifier of the existing owner of the site group and the value of **@CurrentUserIsOwner** is "1". This parameter can be NULL. If this parameter is NULL, the update MUST fail and the return code MUST be 80. However, if the user does not have sufficient permissions, the return code MUST be 5.

@DLAlias: The new email address for the distribution list associated with this site group. This parameter can be NULL.

@DLErrorMessage: The most recent error message returned by an asynchronous email distribution list operation, if any. This parameter can be NULL.

@DLFlags: An integer containing new status flags for the email distribution list associated with this group, as defined for the **DLFlags** column in the Sec_SiteGroupsView (section 2.2.7.6). This parameter can be NULL.

@DLJobId: Contains the job ID of a new currently pending asynchronous distribution list operation, or 0 if there is no pending operation. This parameter can be NULL.

@DLArchives: An array of bytes containing a new list of pairs of List Identifiers (section 2.2.1.5) defining additional Lists, which are the recipients of email sent to the email distribution list via the address in **@DLAlias**. Each GUID MUST be stored as a string, with commas separating the List's parent Site GUID and the List GUID, and with semicolons following each pair. This parameter can be NULL.

@RequestEmail: The new email address used to send a request to join or depart this site group. This parameter can be NULL.

@Flags: Contains the new membership permissions bit flags for the site group. The bit flags are defined in the description of the Flags column of the Sec_SiteGroupsView section.

Return Values: The **proc_SecSetSiteGroupProperties** stored procedure MUST return an integer return code from the following table**:**

Copyright © 2014 Microsoft Corporation.

Value	Description
0	Successful execution.
5	The current user does not have sufficient permissions to manage the site group, or the site collection is set to read-only.
80	An update error occurred or a parameter was invalid.

The **proc_SecSetSiteGroupProperties** stored procedure MUST NOT return a result set.

3.1.5.108 proc_SecSetUserAccountDirectoryPath

The **proc_SecSetUserAccountDirectoryPath** stored procedure is invoked to set or clear the User Account Directory Path for a specified site collection.

```
PROCEDURE proc_SecGetUserAccountDirectoryPath(
      @SiteId                       uniqueidentifier,
      @Restriction                  nvarchar(512)
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection.

@Restriction: Specifies the User Account Directory Path for the specified site collection. If **@Restriction** is not NULL the **proc_SecSetUserAccountDirectoryPath** stored procedure MUST set the User Account Directory Path for the site collection, and also set the 0x00080000 bit of the Site Collection Flags to 1. If **@Restriction** is NULL, the **proc_SecSetUserAccountDirectoryPath** stored procedure MUST set the User Account Directory Path for the site collection to NULL, and also set the 0x00080000 bit of the Site Collection Flags to 0.

Return Values: The **proc_SecSetUserAccountDirectoryPath** stored procedure returns an integer which MUST be 0.

The **proc_SecGetUserAccountDirectoryPath** stored procedure MUST NOT return a result set.

3.1.5.109 proc_SecSetWebRequestAccess

The **proc_SecSetWebRequestAccess** stored procedure is invoked to set the request access email address for a specified site. WSS routes site access requests to the specified request access email address.

```
PROCEDURE proc_SecSetWebRequestAccess(
      @WebId                        uniqueidentifier,
      @RequestAccessEmail           nvarchar(255)
);
```

@WebId: The Site Identifier (section 2.2.1.11) of the Site to set the request access email address.

@RequestAccessEmail: A string of the new request access email address.

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

The **proc_SecSetWebRequestAccess** stored procedure MUST NOT return a result set.

Copyright © 2014 Microsoft Corporation.

3.1.5.110 proc_SecUpdateAnonymousPermMask

The **proc_SecUpdateAnonymousPermMask** stored procedure is invoked to modify the permissions for the anonymous user. The site (2) MUST have a unique scope identified by **@**ScopeId rather than an inherited scope. If the scope specified by **@**SiteId, **@**WebId, **@**ScopeId does not exist, then no change will occur.

```
PROCEDURE proc_SecUpdateAnonymousPermMask(
      @SiteId                       uniqueidentifier,
      @WebId                        uniqueidentifier,
      @ScopeId                      uniqueidentifier,
      @AnonymousPermMask            bigint
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the scope.

@WebId: The Site Identifier (section 2.2.1.11) of the site (2) associated with the scope.

@ScopeId: The Scope Identifier (section 2.2.1.8) of the scope containing the permission set.

@AnonymousPermMask: A WSS Rights Mask (section 2.2.2.13). The scope's permissions for anonymous users MUST be updated to this value.

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

The **proc_SecUpdateAnonymousPermMask** stored procedure MUST NOT return a result set.

3.1.5.111 proc_SecUpdateDomainGroupMapData

The **proc_SecUpdateDomainGroupMapData** stored procedure is invoked to update the domain group map cache of a specified site collection.

```
PROCEDURE proc_SecUpdateDomainGroupMapData(
      @SiteId                       uniqueidentifier,
      @CacheVersion                 bigint,
      @DomainGroupMapCache          image
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection to update. This parameter MUST correspond to a valid site collection, and MUST NOT be NULL.

@CacheVersion: A signed 64-bit (8-byte) integer containing the cache version number. This value is derived from the value of the SecurityVersion field (2) contained in the site collection Access Control List (ACL). See the WSS ACL (section 2.2.4.6). This parameter MUST NOT be NULL.

@DomainGroupMapCache: The domain group map image is defined in the WSS External Group Map Cache Format (section 2.2.4.7).

Value	Bytes	Description
CachedVersion	8	The current cached security version. This is the same value as @CacheVersion.
DGCount	4	Count of Domain Groups.

Copyright © 2014 Microsoft Corporation.

Value	Bytes	Description
REPEAT		Repeat the following fields "DGCount" times.
DGSigSize	4	Size of the domain group signature, in bytes.
DGSig	DGSigSize	Value of the domain group signature.
PIDCount	4	Count of PIDs (principal IDs) in this domain group.
PIDs	4*PIDCount	The value of each PID.

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

The **proc_SecUpdateDomainGroupMapData** stored procedure MUST NOT return a result set.

3.1.5.112 proc_SecUpdateRoleDef

The **proc_SecUpdateRoleDef** stored procedure is invoked to update a role definition with a new title, description, display order, Role Definition Type (section 2.2.3.16), and WSS Rights Mask (section 2.2.2.13).

```
PROCEDURE proc_SecUpdateRoleDef(
      @SiteId                     uniqueidentifier,
      @WebId                      uniqueidentifier,
      @RoleId                     int,
      @Title                      nvarchar(255),
      @Description                nvarchar(512),
      @Order                      int,
      @Type                       tinyint,
      @PermMask                   bigint
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the role definition to be updated.

@WebId: The Site Identifier (section 2.2.1.11) of the site (2) containing the role definition to be updated.

@RoleId: The Role Identifier (section 2.2.1.7) of the role definition to be updated.

@Title: The title of the role definition for display in the WFE. This MUST be a value that does not match an existing role definition title within the Site, and MUST NOT be NULL.

@Description: The description of the role definition for display in the WFE.

@Order: An integer value specifying the relative position in which this role definition is displayed in the WFE. Multiple role definitions can have the same **@Order** value. This value MUST NOT be NULL.

@Type: The Role Definition Type value for the updated role.

@PermMask: A WSS Rights Mask that specifies the rights to grant to the role definition. If this field (2) is not NULL, the WSS Rights Mask associated with the role definition MUST be set to this value and any permissions associated with this role definition in the Site and site collection MUST be

Copyright © 2014 Microsoft Corporation.

modified. If this field (2) is NULL, the existing WSS Rights Mask associated with the role definition MUST NOT be changed.

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

Value	Description
0	Successful execution.
80	The role definition could not be updated.

Upon successful execution, the **proc_SecUpdateRoleDef** stored procedure MUST return a single result set as follows.

3.1.5.112.1 Site Audit Mask Result Set

The Site Audit Mask Result Set contains information about the Audit Flags (section 2.2.2.1) associated with the Site. If execution is successful, the Site Audit Mask Result Set MUST be returned, and MUST contain one row. The Site Audit Mask Result Set is defined in the Common Result Sets Site Audit Mask Result Set (section 2.2.5.20).

3.1.5.113 proc_SecUpdateUser

The **proc_SecUpdateUser** stored procedure is invoked to update the display name, email address, notes, or administrative privileges listed in the user information associated with a principal (1).

```
PROCEDURE proc_SecUpdateUser(
      @SiteId                       uniqueidentifier,
      @CurrentUserId                int,
      @UserIdToUpdate               int,
      @Title                        nvarchar(255),
      @Email                        nvarchar(255),
      @Notes                        nvarchar(1023),
      @SiteAdmin                    bit
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing both the current user and the principal (1) to be updated.

@CurrentUserId: The User Identifier (section 2.2.1.13) of the current user making the update request.

@UserIdToUpdate: The User Identifier of the principal (1) whose User information is to be updated.

@Title: The display name to be set in the principal's User information. If this parameter is NULL, the User information update MUST fail.

@Email: The email address to be set in the principal's User information. If this parameter is NULL, the User information update MUST fail.

@Notes: The notes text to be set in the principal's User information. If this parameter is NULL, the User information update MUST fail.

@SiteAdmin: A bit specifying whether or not to set the principal (1) as a site administrator.

Copyright © 2014 Microsoft Corporation.

- When this parameter is set to "1", if the current user is a site administrator, and the principal (1) is not a domain group, the principal MUST be set as a site administrator.
- When the **@SiteAdmin** parameter is set to "0", **proc_SecUpdateUser** MUST clear the site administrator flag in the User information for the principal (1).
- If the parameter is NULL, **proc_SecUpdateUser** MUST make no changes to the site administrator flag.

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

Value	Description
0	Successful execution.
1399	A cannot be a site administrator.
4335	Cannot successfully remove the site administrator privilege from the principal (1), as no other site administrator was found.

The **proc_SecUpdateUser** stored procedure MUST return no result sets.

3.1.5.114 proc_SecUpdateUserActiveStatus

The **proc_SecUpdateUserActiveStatus** stored procedure marks a user as an active user in a site collection.

```
PROCEDURE proc_SecUpdateUserActiveStatus(
      @SiteId                       uniqueidentifier,
      @UserId                       int
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the User.

@UserId: The User Identifier (section 2.2.1.13) for the User to be marked as active user. If the User is not found or if the User is the system account, **proc_SecUpdateUserActiveStatus** MUST take no action.

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

The **proc_SecUpdateUserActiveStatus** stored procedure MUST return no result sets.

3.1.5.115 proc_TakeOverCheckOut

The **proc_TakeOverCheckOut** stored procedure is invoked to override checkout for a document that is checked out and has no checked-in draft or published version.

```
PROCEDURE proc_TakeOverCheckOut(
      @SiteId                       uniqueidentifier,
      @ListId                       uniqueidentifier,
      @DirName                      nvarchar(256),
      @LeafName                     nvarchar(128),
      @UserId                       int
```


Copyright © 2014 Microsoft Corporation.

```
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the document.

@ListId: The List Identifier (section 2.2.1.5) of the list (1) containing the document.

@DirName: The directory name of the document.

@LeafName: The leaf name of the document.

@UserId: The User Identifier (section 2.2.1.13) for the user being assigned as the check-out User for the document. This value MUST refer to an existing User Identifier for the specified site collection.

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

Value	Description
0	Successful execution , or the User specified by @UserId does not exist, or specified document is already checked out to specified User.
2	File Not Found: The specified document was not found; or the document is not checked out; or the document is checked out, but another current version of the document exists.

The **proc_TakeOverCheckOut** stored procedure MUST NOT return any result sets.

3.1.5.116 proc_UncheckoutDocument

The **proc_UncheckoutDocument** stored procedure is invoked to release the short-term lock on a document, or to release a checked out document. The current document is set to the most recent previous published or draft version, or to the version currently checked out, depending on the document state and **proc_UncheckoutDocument**'s parameters.

```
PROCEDURE proc_UncheckoutDocument (
      @SiteId                       uniqueidentifier,
      @WebId                        uniqueidentifier,
      @DirName                      nvarchar(256),
      @LeafName                     nvarchar(128),
      @UserId                       int,
      @ShortTerm                    bit,
      @Force                        bit
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the document to have its short-term lock released or its check out reverted.

@WebId: The Site Identifier (section 2.2.1.11) of the site (2) containing the document.

@DirName: The Document's directory name.

@LeafName: The Document's leaf name.

@UserId: The user identifier of the current user.

Copyright © 2014 Microsoft Corporation.

@ShortTerm: Specifies whether to release a short-term lock on the document or to revert a check out of the document. This parameter MUST NOT be NULL.

- If **@ShortTerm** is set to "1" and either the current user holds a short-term lock on the document or **@Force** is set to "1," the lock MUST be released.
- If **@ShortTerm** is set to "0" and either the current user has checked out the document or **@Force** is set to "1," the **check out** MUST be reverted, **proc_UncheckoutDocument** MUST discard any draft document created for the check out and MUST set the most recent previous draft or published version as the current version.

@Force: Specifies whether to force the release of a short-term lock or the reversion of a check out of the document held by another User. If this parameter is set to "1", the short-term lock or checkout of the document (as specified by the **@ShortTerm** parameter) MUST be released, and the most recent previous draft or published version MUST be reverted to the current version, even if the current user is not the holder of the short-term lock or the User the document is checked out to. The owner of the short term lock or checkout can also force the release of a short-term lock or the reversion of a checkout. If **@Force** is not set to "1", the short-term lock or check-out of the document will not be released, and the most recent previous draft or published version will not revert to the current version if the current user is not the holder of the short-term lock or the User the document is checked out to.

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

Value	Description
0	Successful execution.
3	The document was not found at the specified location.
158	The document is not checked out, or does not have a short-term lock, or the current user attempted to release a short-term lock on a document that is checked out, or to revert the check out of a document that has a short-term lock.
173	Another User has checked out the document or has a short-term lock on the document, and @Force is not set to "1".
2401	A checked-out document was specified and no published or draft version of the document exists to revert to.

The **proc_UncheckoutDocument** stored procedure MUST return zero result sets if the return code is 2401. Otherwise **proc_UncheckoutDocument** MUST return two or three result sets in the order specified in the following sections.

3.1.5.116.1 Link Info Result Set

The Link Info Result Set returns information about all forward links and backward links associated with the document. The Link Info Result Set MUST be returned and MUST contain one row for each forward links and backward link associated with the specified document after the short-term lock has been released or the checkout has been reverted.

The Link Info Result Set is defined in the Common Result Sets Link Information Result Set (section 2.2.5.11).

Copyright © 2014 Microsoft Corporation.

3.1.5.116.2 Document Metadata Result Set

The Document Metadata Result Set returns the metadata for the specified document after the short-term lock has been released or the checkout has been reverted. The Document Metadata Result Set MUST be returned and MUST contain one row for the specified document.

The Document Metadata Result Set is defined in the Common Result Sets Document Metadata Result Set (section 2.2.5.6).

3.1.5.116.3 NULL Result Set

The NULL Result Set returns no data. The NULL Result Set MUST only be returned if the Document Metadata Result Set is not empty and **@**WebId is NULL. The NULL Result Set MUST return zero rows in a schema containing a single NULL column.

3.1.5.116.4 Event Receivers Result Set

The Event Receivers Result Set contains information about the list item event receivers defined for this document.

The Event Receivers Result Set MUST only be returned if the Document Metadata Result Set is not empty and **@WebId** is not NULL. There MUST be one row for each event receiver with an Event Host Type (section 2.2.3.5) of 3 (list item) registered for the specified document after the short-term lock has been released or the checkout has been reverted.

The Event Receivers Result Set is defined in the Common Result Sets Event Receivers Result Set (section 2.2.5.9).

3.1.5.117 proc_UpdateDocBuildDependencySet

The **proc_UpdateDocBuildDependencySet** stored procedure is invoked to store an updated version of a build dependency set for a specified document.

```
PROCEDURE[proc_UpdateDocBuildDependencySet](
      @DocSiteId                    uniqueidentifier,
      @DocDirName                   nvarchar(256),
      @DocLeafName                  varchar(128),
      @Level                        tinyint,
      @BuildDependencySet           image
);
```

@DocSiteID: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the document.

@DocDirName: The directory name of the document.

@DocLeafName: The leaf name of the document.

@Level: The Publishing Level Type (section 2.2.2.6) value of the document containing the dependency set to update.

@BuildDependencySet: An image containing the build dependency set for the document. This parameter MUST NOT be NULL. **Use proc_DeleteDocBuildDependencySet** to set the document dependencies to NULL.

Copyright © 2014 Microsoft Corporation.

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

The **proc_UpdateDocBuildDependencySet** stored procedure MUST NOT return a result set.

3.1.5.118 proc_UpdateDocument

The **proc_UpdateDocument** stored procedure is invoked to update the metadata and contents of a document.

```
PROCEDURE [proc_UpdateDocument](
      @DocSiteId                    uniqueidentifier,
      @DocWebId                     uniqueidentifier,
      @DocDirName                   nvarchar(256),
      @DocLeafName                  nvarchar(128),
      @DocContent                   image,
      @DocMetaInfo                  image,
      @DocSize                      int,
      @DocMetainfoSize              int,
      @DocDirty                     bit,
      @DocVersion                   int,
      @DocFlags                     int,
      @DocIncomingCreatedDTM        datetime,
      @DocIncomingDTM               datetime,
      @ThicketMainFile              bit,
      @GetWebListForNormalization   bit,
      @VersioningEnabled            bit,
      @EnableMinorVersions          bit,
      @UserId                       int,
      @AttachmentOp                 int,
      @PutFlags                     int,
      @UpdateFlags                  int,
      @CharSet                      int,
      @ProgId                       nvarchar(255),
      @AuditMask                    bit,
      @WebParts                     bit,
      @Comment                      nvarchar(1023),
      @IsForceCheckout              bit,
      @IsModerate                   bit,
      @NeedsDraftOwnerRestriction   bit,
      @WelcomePageUrl               nvarchar(260),
      @WelcomePageParameters        ntext,
      @VirusVendorID                int,
      @VirusStatus                  int,
      @VirusInfo                    nvarchar(255),
      @LockTimeout                  int,
      @RefreshLock                  bit,
      @@DocId                       uniqueidentifier            OUTPUT,
      @@Level                       tinyint                     OUTPUT,
      @@DoclibRowId                 int                         OUTPUT,
      @@DocDTM                      datetime                    OUTPUT,
      @@DocUIVersion                int                         OUTPUT,
      @@DocFlagsOut                 int                         OUTPUT,
      @ChunkSize                    int,
      @@DocTextptr                  varbinary(16)               OUTPUT
);
```


Copyright © 2014 Microsoft Corporation.

@DocSiteId: The Site Collection Identifier (section 2.2.1.9) for the site collection containing the document to be updated.

@DocWebId: The Site Identifier (section 2.2.1.11) for the site (2) containing the document.

@DocDirName: The directory name of the document location.

@DocLeafName: The leaf name of the document location.

@DocContent: The document stream of the content of the document.

@DocMetaInfo: document metadata stream for the document.

@DocSize: Size in bytes of the document.

@DocMetainfoSize: Size in bytes of the document's metadata.

@DocDirty: A bit flag specifying whether the document has dependencies, such as links to other items, which need to be updated. If this parameter is set to "1", the document has dependencies to be updated. This flag MUST be stored by the BEDS for subsequent updating of the dependent information.

@DocVersion: The original internal version counter value of the document to be updated. This value is incremented whenever a change is made to this document, and is used for internal conflict detection. If this value does not match the current internal version counter value of the document, **proc_UpdateDocument** MUST NOT update the document version. If the current version number is NULL or the **@PutOpt_PreserveItemVersion** flag is set to 1, **proc_UpdateDocument** MUST NOT update the document version.

@DocFlags: A Doc Flags (section 2.2.2.3) value with metadata describing the document to be updated.

@DocIncomingCreatedDTM: A timestamp in UTC format that specifies when the document was created.

@DocIncomingDTM: A timestamp in UTC format of the last modification date/time of the document. This value can be NULL, specifying the current time on the BEDS as the last modification time.

@ThicketMainFile: A bit flag specifying whether the document is a thicket main file. If this parameter is set to "1", the document is a thicket main file and **proc_UpdateDocument** MUST store this information.

@GetWebListForNormalization: A bit flag specifying whether to retrieve the list of Sites whose immediate parent Site is the Site containing the document to be updated. If this parameter is set to "1", the procedure MUST return a list of subsite of the document's containing Site in a Site List For Normalization Result Set (section 3.1.5.118.1) upon successful completion.

@VersioningEnabled: A bit flag specifying whether UI version numbering is enabled on the document. If this parameter is set to "1", major UI version numbering MUST be enabled on the document. This parameter MUST be ignored for ghosted items and documents that are not list items or in a document library.

@EnableMinorVersions: A bit flag specifying whether minor UI version numbering is enabled on the document. If this parameter is set to "1", minor UI version numbering MUST be enabled on the document. This parameter MUST be ignored for ghosted items and documents that are not List Items or in a document library.

Copyright © 2014 Microsoft Corporation.

@UserId: The User Identifier (section 2.2.1.13) of the current user that is making the request to the WFE.

@AttachmentOp: An integer containing Attachments Flags (section 2.2.3.1) describing the document. This parameter can have zero or more flags set and MAY be NULL.

@PutFlags: A Put Flags (section 2.2.2.7) value that specifies document update options. This parameter can have zero or more flags set and can be NULL.

@UpdateFlags: A bit field (2) that tracks additional document change options. The only valid value of the **@UpdateFlags** bits is specified as follows.

Value	Meaning
0x00000001	Delete links no longer applicable after document is updated.

@CharSet: An OPTIONAL identifier for the windows code page to associate with the document. Any windows code page identifier is valid for **@CharSet**. This value can be NULL, specifying no preference.

@ProgId: Specifies an OPTIONAL preferred application to open this document. The ProgId is used to distinguish among different applications that save files with a given file extension (i.e. different editors for HTML or XML files). This value can be NULL, specifying no preference.

@AuditMask: A bit flag specifying whether to return the current Audit Flags (section 2.2.2.1) values for the document. If this parameter is set to "1", an Audit Flags Result Set MUST be returned on successful completion.

@WebParts: A bit flag specifying whether the update to the document includes updates to web parts . If this parameter is set to "1", **proc_UpdateDocument** MUST update the document's links to web parts.

@Comment: A description provided when a document is checked in or published, which could be displayed by a version management UI. This value MUST be ignored for documents that do not exist or that are not checked in or published.

@IsForceCheckout: This parameter is unused and MUST be ignored.

@IsModerate: A bit flag specifying whether moderation is in effect for this document. If this parameter is set to "1", moderation is in effect on this document, and **proc_UpdateDocument** MUST apply moderation rules preventing document publication and implementing an approval process when the document is updated.

@NeedsDraftOwnerRestriction: A bit flag specifying whether the draft owner for this document is the only User allowed to update it. If this parameter is set to "1" and a DraftOwnerId exists for the document, **proc_UpdateDocument** MUST only update the document if the current user is the draft owner.

@WelcomePageUrl: If this document is a folder, this specifies a welcome page to redirect to when the folder is requested with an HTTP GET operation. This value can be NULL, specifying no redirection.

@WelcomePageParameters: Specifies query string or bookmark parameters, if any, to append to the URI specified by **@WelcomePageUrl**. This value can be NULL, specifying no parameters to add.

@VirusVendorID: If provided, specifies the vendor identifier of the virus scanner that processed this document.

Copyright © 2014 Microsoft Corporation.

@VirusStatus: A Virus Status (section 2.2.3.18) value specifying the current virus check status of this document.

@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 the document does not exist or if the document has not been processed by a virus scanner.

@LockTimeout: An integer value specifying the number of minutes short-term locking is to be maintained in effect for the document. A value of 0 means the short-term lock MUST be released immediately. **@LockTimeout** MUST be NULL if no short-term lock is being applied, refreshed, or released on the document.

@RefreshLock: Specifies whether to refresh a short-term lock on the document.

- If this parameter is set to "1", the existing short-term lock on the document MUST be set to remain for the number of minutes specified by **@LockTimeout**.
- If this parameter is "0" then the time remaining for an existing short-term lock on the document MUST remain unchanged.
- If there is no short-term lock on the document, this parameter MUST be NULL.

@@DocId: Output parameter. The document identifier (section 2.2.1.2) of the document.

@@Level: Output parameter. A Publishing Level Type (section 2.2.2.6) value indicating the maximum publishing level of the document to be returned to the WFE if multiple publishing levels of the document are available.

@@DoclibRowId: Output parameter. The identifier of the row representing this document in the document library. This value MUST be 0 if the document is not stored in a List.

@@DocDTM: Output parameter. The timestamp of the last modification date of the document. Set to the value of **@DocIncomingDTM**, or to the current UTC date if **@DocIncomingDTM** is NULL.

@@DocUIVersion: Output parameter. The UI version number associated with this document. This value MUST be NULL in the case of a document that does not exist.

@@DocFlagsOut: Output parameter. A Doc Flags value, set to the DocFlags describing this document after successful completion of the update.

@ChunkSize: Specifies the size in bytes that the document MUST be in order for the content to be included in the Update query. This value can be NULL if the **@DocContent** parameter is NULL.

@@DocTextptr: Output parameter. A varbinary pointer set to the value of **@DocContent** if **@DocSize** is greater than **@ChunkSize**. Otherwise, **@DocTextptr** is set to NULL.

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

Value	Description
0	Successful execution.
3	The document does not exist or has been deleted.
5	The User accessing the document is not the owner of the draft, and ownership restriction of the draft is being enforced.

Copyright © 2014 Microsoft Corporation.

Value	Description
33	Locking violation or mismatch of @RefreshLock and short-term lock status before lock timeout.
80	This document does not have a document stream, but a content file exists.
87	The list item for this document could not be added.
154	The Minor UI version number is too high.
158	Checkout is enforced, but the document is not checked out.
160	Document is at draft Level and @UserId is not permitted to see Drafts or is not the draft owner.
173	User does not have appropriate permissions for the type of checkout desired.
212	The document is locked (either not the current version, and is currently checked out, or not enough lock quota); the document cannot be updated.
288	The document is not the current version; the current user is not the owner; the document cannot be updated.
1150	Version mismatch with @DocVersion prevented document update.
1630	Unsupported File Type.
1816	Not enough quota for locking; the document cannot be updated.

The **proc_UpdateDocument** stored procedure can return zero to three result sets in the order listed.

3.1.5.118.1 Site List For Normalization Result Set

The Site List For Normalization Result Set returns a list of URLs for the immediate child subsites of the Site containing the newly added document. When the input parameter **@GetWebListForNormalization** is set to "1" and execution has been successful up to the point of updating the document, the Site List For Normalization Result Set MUST be produced. The Site List For Normalization Result Set MUST contain one row for each subsite found.

The Site List For Normalization Result Set is defined in the URL Result Set (section 2.2.5.26).

3.1.5.118.2 Site Audit Mask Result Set

The Site Audit Mask Result Set returns audit configuration information. When the input parameter **@AuditMask** is set to "1", the Site Audit Mask Result Set MUST be returned, and MUST contain only one row. The Site Audit Mask Result Set is defined in the Common Result Sets Site Audit Mask Result Set (section 2.2.5.20).

3.1.5.118.3 Lock Information Result Set

The Lock Information Result Set returns short-term lock information for the document. The Lock Information Result Set MUST be returned when a successful document update has been completed, and a document has either been checked in or remains checked out by the system. The Lock Information Result Set MUST NOT be returned if the document has not been successfully updated or @LockTimeout is NULL. The Lock Information Result Set MUST contain only one row.

```
tp_Login                    nvarchar(255),
```


Copyright © 2014 Microsoft Corporation.

```
      CheckoutDate                  datetime,
      {CheckoutExpires}             datetime;
```

tp_Login: The login name of the principal (1) to whom the document is checked out. If the document is currently checked in, this MUST be NULL.

CheckoutDate: A timestamp in UTC format indicating when this document was checked out. If the document is currently checked in, this MUST be NULL.

{CheckoutExpires}: A timestamp in UTC format indicating when the short term lock for this document will expire. If the document is currently checked in or has a long term checkout, this MUST be NULL.

3.1.5.119 proc_UpdateListItem

The **proc_UpdateListItem** stored procedure is invoked to update a list item in a document library or list (1).

```
PROCEDURE proc_UpdateListItem(
      @SiteId                       uniqueidentifier,
      @WebId                        uniqueidentifier,
      @ListId                       uniqueidentifier,
      @ItemId                       int,
      @RowOrdinal                   int,
      @Size                         int,
      @ExtraItemSize                int               = NULL,
      @ItemName                     nvarchar(255)     = NULL,
      @UseNvarchar1ItemName         bit               = 1,
      @ItemDirName                  nvarchar(256)     = NULL     OUTPUT,
      @ItemLeafName                 nvarchar(256)     = NULL     OUTPUT,
      @UserId                       int,
      @Level                        tinyint           = 1        OUTPUT,
      @TimeNow                      datetime,
      @NeedsAuthorRestriction       bit               = 0,
      @NeedsDraftOwnerRestriction   bit               = 0,
      @PreserveVersion              bit               = 0,
      @IsMeetingsList               bit               = NULL,
      @IsIssueList                  bit               = NULL,
      @IsNotUserDisplayed           bit               = NULL,
      @SystemUpdate                 bit               = 0,
      @ChangeLevel                  bit               = 0,
      @CheckinItem                  bit               = 0,
      @NeedClone                    bit               = 0,
      @MajorVersionsLimit           int               = 0,
      @MajorMinorVersionsLimit      int               = 0,
      @IsDocLib                     bit               = 0,
      @CheckSchemaVersion           int               = NULL,
      @tp_Ordering                  varchar(512)      = NULL,
      @tp_ThreadIndex               varbinary(512)    = NULL,
      @tp_HasAttachment             bit               = NULL,
      @tp_ModerationStatus          int               = NULL,
      @tp_IsCurrent                 bit               = NULL,
      @tp_ItemOrder                 float             = NULL,
      @tp_Version                   int               = NULL,
      @tp_InstanceID                int               = NULL,
      @tp_ContentType               nvarchar(255)     = NULL,
      @tp_ContentTypeId             varbinary(512)    = NULL,
```


Copyright © 2014 Microsoft Corporation.

```
        @tp_CopySource                  nvarchar(260)       = NULL,
        @tp_HasCopyDestinations         bit                 = NULL,
        @OnRestore                      bit                 = 0,
        @BumpLastDelete                 bit                 = NULL,
        @CreateItemVersion              bit                 = 0,
        @UIVersion                      int                 = NULL,
        @NewUIVersion                   int                 = NULL      OUTPUT,
        @ReturnRowset                   bit                 = 1,
        @tp_Author                      int                 = NULL,
        @tp_Editor                      int                 = NULL,
        @tp_Created                     datetime            = NULL,
        @tp_Modified                    datetime            = NULL,
        @tp_WorkflowVersion             int                 = NULL,
  @nvarchar1!!nvarchar(255)!!= NULL,
        @nvarchar2                      nvarchar(255)       = NULL,
        @nvarchar3                      nvarchar(255)       = NULL,
        @nvarchar4                      nvarchar(255)       = NULL,
        @nvarchar5                      nvarchar(255)       = NULL,
        @nvarchar6                      nvarchar(255)       = NULL,
        @nvarchar7                      nvarchar(255)       = NULL,
        @nvarchar8                      nvarchar(255)       = NULL,
        @nvarchar9                      nvarchar(255)       = NULL,
        @nvarchar10                     nvarchar(255)       = NULL,
        @nvarchar11                     nvarchar(255)       = NULL,
        @nvarchar12                     nvarchar(255)       = NULL,
        @nvarchar13                     nvarchar(255)       = NULL,
        @nvarchar14                     nvarchar(255)       = NULL,
        @nvarchar15                     nvarchar(255)       = NULL,
        @nvarchar16                     nvarchar(255)       = NULL,
        @nvarchar17                     nvarchar(255)       = NULL,
        @nvarchar18                     nvarchar(255)       = NULL,
        @nvarchar19                     nvarchar(255)       = NULL,
        @nvarchar20                     nvarchar(255)       = NULL,
        @nvarchar21                     nvarchar(255)       = NULL,
        @nvarchar22                     nvarchar(255)       = NULL,
        @nvarchar23                     nvarchar(255)       = NULL,
        @nvarchar24                     nvarchar(255)       = NULL,
        @nvarchar25                     nvarchar(255)       = NULL,
        @nvarchar26                     nvarchar(255)       = NULL,
        @nvarchar27                     nvarchar(255)       = NULL,
        @nvarchar28                     nvarchar(255)       = NULL,
        @nvarchar29                     nvarchar(255)       = NULL,
        @nvarchar30                     nvarchar(255)       = NULL,
        @nvarchar31                     nvarchar(255)       = NULL,
        @nvarchar32                     nvarchar(255)       = NULL,
        @nvarchar33                     nvarchar(255)       = NULL,
        @nvarchar34                     nvarchar(255)       = NULL,
        @nvarchar35                     nvarchar(255)       = NULL,
        @nvarchar36                     nvarchar(255)       = NULL,
        @nvarchar37                     nvarchar(255)       = NULL,
        @nvarchar38                     nvarchar(255)       = NULL,
        @nvarchar39                     nvarchar(255)       = NULL,
        @nvarchar40                     nvarchar(255)       = NULL,
        @nvarchar41                     nvarchar(255)       = NULL,
        @nvarchar42                     nvarchar(255)       = NULL,
        @nvarchar43                     nvarchar(255)       = NULL,
        @nvarchar44                     nvarchar(255)       = NULL,
        @nvarchar45                     nvarchar(255)       = NULL,
        @nvarchar46                     nvarchar(255)       = NULL,
```


Copyright © 2014 Microsoft Corporation.

```
@nvarchar47                 nvarchar(255)      = NULL,
@nvarchar48                 nvarchar(255)      = NULL,
@nvarchar49                 nvarchar(255)      = NULL,
@nvarchar50                 nvarchar(255)      = NULL,
@nvarchar51                 nvarchar(255)      = NULL,
@nvarchar52                 nvarchar(255)      = NULL,
@nvarchar53                 nvarchar(255)      = NULL,
@nvarchar54                 nvarchar(255)      = NULL,
@nvarchar55                 nvarchar(255)      = NULL,
@nvarchar56                 nvarchar(255)      = NULL,
@nvarchar57                 nvarchar(255)      = NULL,
@nvarchar58                 nvarchar(255)      = NULL,
@nvarchar59                 nvarchar(255)      = NULL,
@nvarchar60                 nvarchar(255)      = NULL,
@nvarchar61                 nvarchar(255)      = NULL,
@nvarchar62                 nvarchar(255)      = NULL,
@nvarchar63                 nvarchar(255)      = NULL,
@nvarchar64                 nvarchar(255)      = NULL,
@int1                       int                = NULL,
@int2                       int                = NULL,
@int3                       int                = NULL,
@int4                       int                = NULL,
@int5                       int                = NULL,
@int6                       int                = NULL,
@int7                       int                = NULL,
@int8                       int                = NULL,
@int9                       int                = NULL,
@int10                      int                = NULL,
@int11                      int                = NULL,
@int12                      int                = NULL,
@int13                      int                = NULL,
@int14                      int                = NULL,
@int15                      int                = NULL,
@int16                      int                = NULL,
@float1                     float              = NULL,
@float2                     float              = NULL,
@float3                     float              = NULL,
@float4                     float              = NULL,
@float5                     float              = NULL,
@float6                     float              = NULL,
@float7                     float              = NULL,
@float8                     float              = NULL,
@float9                     float              = NULL,
@float10                    float              = NULL,
@float11                    float              = NULL,
@float12                    float              = NULL,
@datetime1                  datetime           = NULL,
@datetime2                  datetime           = NULL,
@datetime3                  datetime           = NULL,
@datetime4                  datetime           = NULL,
@datetime5                  datetime           = NULL,
@datetime6                  datetime           = NULL,
@datetime7                  datetime           = NULL,
@datetime8                  datetime           = NULL,
@bit1                       bit                = NULL,
@bit2                       bit                = NULL,
@bit3                       bit                = NULL,
@bit4                       bit                = NULL,
@bit5                       bit                = NULL,
```


Copyright © 2014 Microsoft Corporation.

```
@bit6                       bit               = NULL,
@bit7                       bit               = NULL,
@bit8                       bit               = NULL,
@bit9                       bit               = NULL,
@bit10                      bit               = NULL,
@bit11                      bit               = NULL,
@bit12                      bit               = NULL,
@bit13                      bit               = NULL,
@bit14                      bit               = NULL,
@bit15                      bit               = NULL,
@bit16                      bit               = NULL,
@uniqueidentifier1          uniqueidentifier  = NULL,
@ntext1                     ntext             = NULL,
@ntext2                     ntext             = NULL,
@ntext3                     ntext             = NULL,
@ntext4                     ntext             = NULL,
@ntext5                     ntext             = NULL,
@ntext6                     ntext             = NULL,
@ntext7                     ntext             = NULL,
@ntext8                     ntext             = NULL,
@ntext9                     ntext             = NULL,
@ntext10                    ntext             = NULL,
@ntext11                    ntext             = NULL,
@ntext12                    ntext             = NULL,
@ntext13                    ntext             = NULL,
@ntext14                    ntext             = NULL,
@ntext15                    ntext             = NULL,
@ntext16                    ntext             = NULL,
@ntext17                    ntext             = NULL,
@ntext18                    ntext             = NULL,
@ntext19                    ntext             = NULL,
@ntext20                    ntext             = NULL,
@ntext21                    ntext             = NULL,
@ntext22                    ntext             = NULL,
@ntext23                    ntext             = NULL,
@ntext24                    ntext             = NULL,
@ntext25                    ntext             = NULL,
@ntext26                    ntext             = NULL,
@ntext27                    ntext             = NULL,
@ntext28                    ntext             = NULL,
@ntext29                    ntext             = NULL,
@ntext30                    ntext             = NULL,
@ntext31                    ntext             = NULL,
@ntext32                    ntext             = NULL,
@sql_variant1               sql_variant       = NULL,
@error_sql_variant1         int               = 0,
@sql_variant2               sql_variant       = NULL,
@error_sql_variant2         int               = 0,
@sql_variant3               sql_variant       = NULL,
@error_sql_variant3         int               = 0,
@sql_variant4               sql_variant       = NULL,
@error_sql_variant4         int               = 0,
@sql_variant5               sql_variant       = NULL,
@error_sql_variant5         int               = 0,
@sql_variant6               sql_variant       = NULL,
@error_sql_variant6         int               = 0,
@sql_variant7               sql_variant       = NULL,
@error_sql_variant7         int               = 0,
@sql_variant8               sql_variant       = NULL,
```


Copyright © 2014 Microsoft Corporation.

```
        @error_sql_variant8             int                 = 0,
        @eventData                      image               = NULL,
        @acl                            image               = NULL
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) for the site collection containing the list item to be updated.

@WebId: The Site Identifier (section 2.2.1.11) for the site (2) containing the list item.

@ListId: The List Identifier (section 2.2.1.5) of the List containing the list item.

@ItemId: The integer identifier of the list item.

@RowOrdinal: The 0-based ordinal of the row to update in the set of rows representing this list item in the AllUserData (section 2.2.7.3) table. If a list item requires multiple rows to represent it in the AllUserData table because it contains more defined data columns than will fit in a single row, this parameter specifies which row to update with this call. If more than one row of data for the list item is to be updated, the WFE MUST call **proc_UpdateListItem** once for each updated row. This parameter MUST NOT be NULL.

@Size: The new size, in bytes, of the list item row to be updated. MUST NOT be NULL.

@ExtraItemSize: The size of the predefined SQL parameter fields in the list item row.

@ItemName: The new display name for the list item.

@UseNvarchar1ItemName: If **@ItemName** is NULL, this bit flag specifies whether to use the content of **@nvarchar1** for the new display name for the list item.

@ItemDirName: An output parameter containing the directory name of the updated list item.

@ItemLeafName: An output parameter containing the leaf name of the updated list item.

@UserId: The User Identifier (section 2.2.1.13) for the current user. **proc_UpdateListItem** uses this parameter for comparison when the @NeedsAuthorRestriction or **@NeedsDraftOwnerRestriction** parameters are set. If this parameter is NULL, **proc_UpdateListItem** MUST use the User Identifier stored as the List Item's editor as the current user.

@Level: A Publishing Level Type (section 2.2.2.6) value specifying the publishing status of the updated list item.

@TimeNow: The current time in UTC format on the BEDS. This parameter MUST NOT be NULL.

@NeedsAuthorRestriction: A bit flag specifying whether only the List Item's author is permitted to update the list item. If this parameter is set to 1, the current user MUST be the List Item's author for successful execution.

@NeedsDraftOwnerRestriction: A bit flag specifying whether only the List Item's draft owner is permitted to update the list item. If this parameter is set to 1, the current user MUST be the List Item's draft owner for successful execution.

@PreserveVersion: A bit flag specifying whether to preserve the internal version number of the list item to be updated. If this parameter is set to 1, the internal version number of the list item MUST NOT be incremented by **proc_UpdateListItem** as part of the update.

Copyright © 2014 Microsoft Corporation.

@IsMeetingsList: A bit flag specifying whether the list item is contained in a Meetings List (a List with a List Server Template (section 2.2.3.12) value of 200).

@IsIssueList: An OPTIONAL bit flag specifying whether the list item is contained in an Issues List (a List with a List Server Template value of 1100). This parameter MUST be ignored.

@IsNotUserDisplayed: A bit flag specifying whether the display name of the current user is not to be displayed in logging events for this update. If this parameter is set to 1, the display name of the current user MUST NOT be used when **proc_UpdateListItem** logs events due to the update, and the string "***" MUST be used instead.

@SystemUpdate: A bit flag specifying whether to leave the modification time and the editor fields unchanged during this update. If this parameter is set to 1, **proc_UpdateListItem** MUST update the database with changes made to the list item without effecting changes in the modification time or editor fields.

@ChangeLevel: A bit flag specifying whether to recalculate the publishing level of the list item as part of the update. If this parameter is set to 1, **proc_UpdateListItem** MUST recalculate and set the updated List Item's publishing level.

@CheckinItem: A bit flag specifying whether to set the publishing level for the updated list item to published. If this parameter is set to 1, **proc_UpdateListItem** MUST set the updated List Item's publishing level to 1 (published).

@NeedClone: A bit flag specifying whether a copy of the list item as a draft version is made as part of this update. If this parameter is set to 1, the publishing level specified by the **@Level** parameter MUST also be set to 1 (published) for successful completion.

@MajorVersionsLimit: The number of major versions to keep of the list item.

@MajorMinorVersionsLimit: The number of major and minor versions to keep of the list item.

@IsDocLib: A bit flag specifying whether the list item to be updated is contained within a document library. If this parameter is set to 1, the List specified by **@ListId** MUST be a document library.

@CheckSchemaVersion: This specifies an OPTIONAL schema version number to compare with the List schema version number. If this parameter is not NULL, the version numbers MUST match for successful completion.

@tp_Ordering: This specifies the threading structure for this list item in a legacy Discussion Board List (a List with a List Base Type (section 2.2.3.11) of 3) as a concatenation of timestamp values in yyyyMMddHHmmss format. For all List Items in Lists with other List Base Types, this parameter MUST be NULL.

@tp_ThreadIndex: This specifies the List Item's position within a threaded Discussion Board List (a List with a List Base Type of 3) as a binary structure. For all List Items in Lists with other List Base Types, this parameter MUST be NULL.

@tp_HasAttachment: A bit flag specifying whether the list item has an associated attachment.

@tp_ModerationStatus: A Moderation Status (section 2.2.3.13) value specifying the current moderation approval status of this list item.

@tp_IsCurrent: A bit flag specifying whether this is the current version of this publishing level of the list item.

Copyright © 2014 Microsoft Corporation.

@tp_ItemOrder: This specifies the relative positioning order to view the list item in when displayed with other List Items from the same List. This value can be the same as other List Items in the List.

@tp_Version: An OPTIONAL value to compare with the internal counter of the list item. If this parameter is not NULL, the parameter MUST match the internal counter for successful completion.

@tp_InstanceID: If this list item is associated with a particular instance of a recurring meeting, this specifies the integer ID of that instance. For all other List Items, this parameter MUST be NULL.

@tp_ContentType: This specifies the name of the content type for this list item.

@tp_ContentTypeId: This specifies the ID of the content type for this list item.

@tp_CopySource: The URL used as a source for this list item. If this list item was not copied from a source list item, this value MUST be NULL.

@tp_HasCopyDestinations: A bit flag specifying whether destination locations have been set for this list item to be copied to. If this list item does not have a destination location set, this value MUST be false.

@OnRestore: A bit flag specifying whether this list item is being inserted by a backup restore operation.

@BumpLastDelete: A bit flag specifying whether to update the **tp_LastDeleted** property on the List specified by **@ListId**. If set to 0, only the **tp_ LastDeleted** property MUST NOT be updated.

@CreateItemVersion: A bit flag specifying whether to create a new version of the list item as part of this update.

@UIVersion: The UI version number to set for the list item.

@NewUIVersion: An output parameter returning the UI version number set for the list item, which MUST be NULL if one was not assigned.

@ReturnRowset: A bit flag specifying whether to return an Item Update Result Set (section 3.1.5.119.1).

@tp_Author: The User Identifier to set as the author of the list item.

@tp_Editor: The User Identifier to set as the last editor of the list item.

@tp_Created: The specified timestamp in UTC format to be set as when this list item was created.

@tp_Modified: The specified timestamp in UTC format to be set as when this list item was last modified.

@tp_WorkflowVersion: If this list item is part of a workflow (2), this specifies the value to set denoting the state of this list item within that workflow (2). If this list item is not part of a workflow (2), this value MUST be NULL.

The next nine columns are duplicated a variable number of times, depending on the List Item's content type within the View definition, with each column referring to a separate List Server Template-defined field (2) or user-defined field within the containing List. Each instance of these individual column names is differentiated by a suffix with a numeric value indicated in the column description, which replaces the placeholder '#' symbol below.

@nvarchar#: User-defined columns in the List containing values of type nvarchar. There are 64 columns numbered from 1 to 64. If the column does not contain data, the value MUST be NULL.

Copyright © 2014 Microsoft Corporation.

@int#: User-defined columns in the List containing values of type int. There are 16 columns numbered from 1 to 16. If the column does not contain data, the value MUST be NULL.

@float#: User-defined columns in the List containing values of type float. There are 12 columns numbered from 1 to 12. If the column does not contain data, the value MUST be NULL.

@datetime#: User-defined columns in the List containing values of type datetime. There are eight columns numbered from 1 to 8. If the column does not contain data, the value MUST be NULL.

@bit#: User-defined columns in the List containing values of type bit. There are 16 columns numbered from 1 to 16. If the column does not contain data, the value MUST be NULL.

@uniqueidentifier1: A user-defined column in the List containing values of type uniqueidentifier. If the column does not contain data, the value MUST be NULL.

@ntext#: User-defined columns in the List containing values of type ntext. There are 32 columns numbered from 1 to 32. If the column does not contain data, the value MUST be NULL.

@sql_variant#: User-defined columns in the List containing values of type sql_variant. There are eight columns numbered from 1 to 8. If the column does not contain data, the value MUST be NULL.

@error_sql_variant#: An integer which specifies the type to be applied to the corresponding values specified as arguments for the parameter **@sql_variant#**. There are eight columns numbered from 1 to 8. The following are valid values:

Value	Description
1	Convert the argument value to a varbinary(2).
2	Convert the argument value to a bit.
3	Convert the argument value to a float.
4	Convert the argument value to a datetime.

@eventData: Contains implementation-specific event data significant to the WFE but otherwise opaque to the BEDS, to be stored by the BEDS for eventual writing to a log file.

@acl: The binary serialization of the WSS ACL Format access control list for the data supplied in **@eventData**, to be stored with the data.

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

Value	Description
0	Successful execution.
3	Could not create a unique filename, or the specified Site does not exist.
5	The current user does not have sufficient permissions to update the list item.
13	The list item to be added is not valid.
87	Unable to update the list item because the input parameters do not match an existing list item, or an error occurred during a table update operation.
160	A valid User Identifier was not specified.

Copyright © 2014 Microsoft Corporation.

Value	Description
212	The database for the Site is locked.
288	The list item cannot be published by the current user because a different User is listed as the draft owner.
1150	Failed to update the list item.
1638	The current schema version of the List does not match the value specified in @CheckSchemaVersion.
1639	Only a published list item can be copied.
1816	The site collection is over its allocated size quota.
4005	The specified list item was not found.

The **proc_UpdateListItem** stored procedure MUST return zero or one result sets as follows:

3.1.5.119.1 Item Update Result Set

The Item Update Result Set returns status information about the list item update. The Item Update Result Set MUST only be returned when the **@**ReturnRowset parameter is set to 1, and MUST contain a single row.

```
{ReturnCode}                int,
{RowsAffected}              int,
{Version}                   int,
{Author}                    int,
{Id}                        uniqueidentifier,
{AuditMask}                 int,
{InheritAuditMask}          int,
{GlobalAuditMask}           int,
{FullUrl}                   nvarchar(260);
```

{ReturnCode}: The return code value returned by **proc_UpdateListItem**.

{RowsAffected}: The count of the AllUserData (section 2.2.7.3) table rows updated by **proc_UpdateListItem**.

{Version}: The current value of the internal counter for the updated list item. This MUST be NULL if the specified list item was not found.

{Author}: The User Identifier (section 2.2.1.13) for the author of the updated list item. This MUST be NULL if the specified list item was not found.

{Id}: The List Item Identifier (section 2.2.1.6) of the updated list item. This MUST be NULL if the specified list item was not found.

{AuditMask}: The Audit Flags (section 2.2.2.1) value for the updated list item. This MUST be NULL if the specified list item was not found.

{InheritAuditMask}: The Audit Flags value for the updated list item inherited from its containing List or document library. This MUST be NULL if the specified list item was not found.

Copyright © 2014 Microsoft Corporation.

{GlobalAuditMask}: The global Audit Flags value for the site collection specified by **@**SiteId. This MUST be NULL if the specified site collection was not found.

{FullUrl}: The store-relative form URL for the updated list item. This MUST be NULL if the specified list item was not found.

3.1.5.120 proc_UpdateListSettings

The **proc_UpdateListSettings** stored procedure is invoked to update list metadata.

```
PROCEDURE proc_UpdateListSettings(
      @SiteId                        uniqueidentifier,
      @WebId                         uniqueidentifier,
      @ListId                        uniqueidentifier,
      @BaseType                      int,
      @ServerTemplate                int,
      @Title                         nvarchar(255),
      @ReadSecurity                  int,
      @WriteSecurity                 int,
      @NewDefaultView                uniqueidentifier,
      @Description                   ntext,
      @TemplateDirName               nvarchar(256),
      @TemplateLeafName              nvarchar(128),
      @Fields                        ntext,
      @Direction                     int,
      @EventSinkAssembly             nvarchar(255),
      @EventSinkClass                nvarchar(255),
      @EventSinkData                 nvarchar(255),
      @Flags                         bigint,
      @ImageUrl                      nvarchar(255),
      @ThumbnailSize                 int,
      @WebImageWidth                 int,
      @WebImageHeight                int,
      @Version                       int,
      @ConvertToGlobalMtgDataList    bit,
      @EmailInsertsFolder            nvarchar(255),
      @EmailInsertsLastSyncTime      nvarchar(50),
      @EmailAlias                    nvarchar(128),
      @SendToLocation                nvarchar(512),
      @NavBarEid                     int,
      @AddOrRemoveFromNavBar         bit,
      @UseRootFolderForNav           bit,
      @MajorVersionCount             int,
      @MajorMinorVersionCount        int,
      @WorkFlowId                    uniqueidentifier,
      @RemoveSystemAlerts            bit,
      @LastItemModifiedDate          datetime)
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the List whose metadata is being updated.

@WebId: The Site Identifier (section 2.2.1.11) of the site (2) containing the List.

@ListId: The List Identifier (section 2.2.1.5) of the List to be updated.

Copyright © 2014 Microsoft Corporation.

@BaseType: A List Base Type (section 2.2.3.11) value. If **@ConvertToGlobalMtgDataList** is equal to "1", this value MUST be used to set the List Base Type; otherwise the value is ignored. This parameter can be NULL.

@ServerTemplate: The identifier for the List Server Template defining the base structure of this List. If **@ConvertToGlobalMtgDataList** is equal to "1", this value MUST be used to set the List Server Template; otherwise the value MUST be ignored.

@Title: User-provided title text for the List, used in the SharePoint UI to identify the List. This parameter MUST NOT be NULL.

@ReadSecurity: Specifies the security policy that MUST be set for read access to list items in the List. Valid read security values are listed in the following table.

Value	Description
1	Users with read permissions can read all List Items.
2	Users with read permissions can only read their own items.
NULL	proc_UpdateListSettings MUST make no changes to read security for the List.

@WriteSecurity: Specifies the security policy that MUST be set for write access on List Items in the List. Valid write security values are listed 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 items only.
4	Users have no write access to any List Items.
NULL	If NULL, proc_UpdateListSettings MUST make no changes to write security for the List.

@NewDefaultView: A View Identifier (section 2.2.1.14) of a defined List view to be displayed by default when navigating to the List. If NULL, **proc_UpdateListSettings** MUST make no changes to the default List view.

@Description: A user-provided readable text description for the List, used in the SharePoint UI. If NULL, **proc_UpdateListSettings** MUST make no changes to the description.

@TemplateDirName: Directory name for the document Template associated with the List. This parameter can be NULL. The parameter MUST be ignored if **@TemplateLeafName** is NULL or contains an empty string.

@TemplateLeafName: Leaf name for the document template associated with the List. If **@TemplateLeafName** is NULL or contains an empty string, **proc_UpdateListSettings** MUST make no changes to this value.

@Fields: Specifies the field (2) definitions for the List, consisting of the concatenation of a WSS implementation-specific version string for the List Template, followed by an XML fragment representation of the field definitions of the List, as specified in the FieldDefinitionDatabaseWithVersion type defined in section 2.2.8.3.5. If this parameter is NULL, **proc_UpdateListSettings** MUST make no changes to the field definitions for the List.

Copyright © 2014 Microsoft Corporation.

@Direction: An enumerated value specifying the direction of text flow for WFE elements presented by this List. If this parameter is NULL, **proc_UpdateListSettings** MUST make no changes to the List direction value. The following are the valid values for **@**Direction.

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

@EventSinkAssembly: A .NET assembly name for an event sink handler for the List. If this parameter is NULL, **proc_UpdateListSettings** MUST make no changes to this value.

@EventSinkClass: A .NET assembly class identifier for an event sink handler for the List. If NULL, **proc_UpdateListSettings** MUST make no changes to this value.

@EventSinkData: Event sink data for an event sink handler for the List. If NULL, **proc_UpdateListSettings** MUST make no changes to this value.

@Flags: A List Flags (section 2.2.2.5) value for the List.

@ImageUrl: A Site-relative URL holding an image associated with the List in the SharePoint UI. If NULL, **proc_UpdateListSettings** MUST make no changes to this value.

@ThumbnailSize: A value specifying the width in pixels used by Lists of type Image Library to determine the rendering size of an image thumbnail. If NULL, **proc_UpdateListSettings** MUST make no changes to this value.

@WebImageWidth: A value specifying the width in pixels used by Lists of type Image Library to determine the rendering width of an image. If NULL, **proc_UpdateListSettings** MUST make no changes to this value.

@WebImageHeight: A value specifying the height in pixels used by Lists of type Image Library to determine the rendering height of an image. If NULL, **proc_UpdateListSettings** MUST make no changes to this value.

@Version: An integer used to match on the internal version counter of the List to be updated. This is used to prevent updates with out-of-date data. On successful execution, the internal version counter value of the List will be incremented by 1.

@ConvertToGlobalMtgDataList: Used by SharePoint to convert List data to use global meeting List data. If this parameter is set to 1, the List data will be converted to global meeting List data; to do this, the List template MUST NOT be 200 (Meeting), 202 (Meeting user) or 212 (Homepage). If set to any other value or NULL, **proc_UpdateListSettings** MUST make no changes to this value.

@EmailInsertsFolder: A URL fragment specifying the directory on the configured email inserts server which SHOULD be inspected for new email messages to be processed for this List. If NULL, **proc_UpdateListSettings** MUST make no changes to this value.

@EmailInsertsLastSyncTime: The time stamp in UTC format when the email inserts server was last processed for new email messages. If NULL, no changes will be made. If this parameter contains an empty string, then the current time will be used.

Copyright © 2014 Microsoft Corporation.

@EmailAlias: The email alias of the List. This alias is used to allow files to be sent directly to this List through an implementation-specific email-handling feature. If this parameter is NULL, **proc_UpdateListSettings** MUST make no changes to this value.

@SendToLocation: A WSS implementation-specific string holding a URL used in the List "Send To Location" feature. If this parameter is NULL, **proc_UpdateListSettings** MUST make no changes to this value.

@NavBarEid: Specifies a navigation node element identifier for the Parent navigation node of the node that will represent the List. If **@**AddOrRemoveFromNavBar is NULL or 0, this parameter MUST be ignored. Otherwise, it MUST contain a valid navigation node entity identifier.

@AddOrRemoveFromNavBar: Specifies whether to add or remove this List from the Site's navigation bar. If this parameter is set to "0", the List MUST be removed from the Site's navigation bar. If NULL, **proc_UpdateListSettings** MUST make no changes to the Site's navigation bar. Otherwise, if the List is not in the Site's navigation bar, it MUST be added to it. To move nodes from one location to another, the navigation node SHOULD be first removed, and then re-added with the new desired setting or else duplicate node entries will occur.

@UseRootFolderForNav: Specifies whether the List's root folder is used for the List entry in the Site's navigation bar, instead of the default view page. If this parameter is set to "1", the root folder of the List MUST be set as the navigation bar target. If set to "0", the default view page for the List MUST be set as the navigation bar target. If the **@AddOrRemoveFromNavBar** parameter is set to NULL, this parameter MUST also be NULL. Otherwise, this parameter can be NULL, which MUST have the effect of being set to "0". To change the root folder navigation from one node to another, the current navigation node SHOULD be first removed, and then the new one added with the desired setting or else duplicate node entries will occur.

@MajorVersionCount: Specifies the maximum number of major versions of List Items to be retained by the List.

@MajorMinorVersionCount: Specifies the maximum number of minor versions of List Items to be retained by the List.

@WorkFlowId: Specifies the Workflow Identifier (section 2.2.1.16) of the default workflow (2) of the List.

@RemoveSystemAlerts: A bit flag specifying whether to remove system alerts for this List from the email subscriptions for the system administration user. If this parameter is set to "1", system alerts on the List will be removed.

@LastItemModifiedDate: A time stamp, in UTC format, that specifies when the List was last modified. If **@**LastItemModifiedDate is NULL, and any of **@Title**, **@Fields**, **@Flags**, or **@Description** is not NULL, **proc_UpdateListSettings** MUST use the current time. If **@LastItemModifiedDate** is NULL, and **@Title**, **@Fields**, **@Flags**, and **@Description** are also NULL, **proc_UpdateListSettings** MUST make no changes to this value.

Return Values: The **proc_UpdateListSettings** stored procedure returns an integer return code that MUST be listed in the following table.

Value	Description
0	Successful execution.
3	The web does not exist.

Copyright © 2014 Microsoft Corporation.

Value	Description
13	@ConvertToGlobalMtgDataList is set to 1, but the List template is not valid for using global data.
15	The template document URL (combination of @TemplateDirName and @TemplateLeafName) is not valid.
52	Duplicate title: a List with an identical title exists.
1150	Concurrency violation occurred.

The **proc_UpdateListSettings** stored procedure MUST return no result sets.

3.1.5.121 proc_UpdateUserInfoInTableFromRowUpdater

The **proc_UpdateUserInfoInTableFromRowUpdater** stored procedure updates the user information data for the specified User.

```
PROCEDURE proc_UpdateUserInfoInTableFromRowUpdater(
      @SiteId                       uniqueidentifier,
      @UserId                       int,
      @Title                        nvarchar(255)      = NULL,
      @Email                        nvarchar(255)      = NULL,
      @IsActive                     bit                = NULL,
      @Locale                       int                = NULL,
      @CalendarType                 smallint           = NULL,
      @AdjustHijriDays              smallint           = NULL,
      @TimeZone                     smallint           = NULL,
      @Collation                    smallint           = NULL,
      @Time24                       bit                = NULL,
      @AltCalendarType              tinyint            = NULL,
      @CalendarViewOptions          tinyint            = NULL,
      @WorkDays                     smallint           = NULL,
      @WorkDayStartHour             smallint           = NULL,
      @WorkDayEndHour               smallint           = NULL
);
```

@SiteId: The Site Collection Identifier (section 2.2.1.9) of the site collection containing the User whose information is to be updated.

@UserId: The User Identifier (section 2.2.1.13) for the User whose information is to be updated.

@Title: The user-friendly display name of the User. If this parameter is NULL, the display name MUST be set to the empty string.

@Email: The email address of the User. If this parameter is NULL, the email address MUST be set to the empty string.

@IsActive: A bit flag specifying whether the User is an active user in the site collection. This flag is set to "0" if the User is not an active user. This flag is set to a non-zero positive integer to indicate otherwise. This flag MUST NOT be NULL.

@Locale: An LCID specifying the preferred locale settings to be used when formatting and displaying UI for the User.

@CalendarType: The Calendar Type (section 2.2.3.3) to be used when processing date values for this User. See the Calendar Type section for details.

Copyright © 2014 Microsoft Corporation.

@AdjustHijriDays: If the **@CalendarType** parameter value is "6", this parameter specifies the number of days to extend or reduce the current month in Hijri calendars for this User.

@TimeZone: The Time Zone Identifier for the time zone to be used when displaying time values for this User. See the Time Zone Identifier (section 2.2.3.17) for details.

@Collation: The Collation Order to be used when displaying information to this User. See the Collation Order (section 2.2.3.4) for details.

@Time24: A bit flag which specifies whether to use a 24-hour time format when displaying time values to this User. If this parameter is set to "1", the 24-hour time format is used; otherwise, the 12-hour time format is used.

@AltCalendarType: The Calendar Type of an alternate calendar for processing date values for this User.

@CalendarViewOptions: A Calendar View Options Type which specifies the calendar display options setting for this User. See the Calendar View Options Type (section 2.2.4.1) for details.

@WorkDays: A set of Workdays Flags which specify the week days defined as the work week for this User. See the Workdays Flags (section 2.2.2.12) for details.

@WorkDayStartHour: The start time of the work day for this User, in minutes from 12**:**00AM. For example, the value 480 indicates 8:00AM.

@WorkDayEndHour: The end time of the work day for this User, in minutes from 12:00AM. For example, the value 1020 indicates 5:00PM.

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

The **proc_UpdateUserInfoInTableFromRowUpdater** stored procedure MUST return no result sets.

3.1.5.122 proc_UrlToWebUrl

The **proc_UrlToWebUrl** stored procedure result set returns the store-relative form URL for the specified site collection which contains the store-relative form URL passed in.

```
PROCEDURE proc_UrlToWebUrl(
      @WebSiteId                    uniqueidentifier,
      @Url                          nvarchar(260)
);
```

@WebSiteId: The Site Collection Identifier (section 2.2.1.9) for a site collection that contains the **@Url** parameter value. The Site Collection Identifier MUST match an existing site collection which could hold the **@Url** parameter value. If the identifier does not match an existing site collection, then an integer value of 1168 is returned along with a blank Web Url result set.

@Url: The complete store-relative form URL, which SHOULD fall under the specified site collection.

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

Copyright © 2014 Microsoft Corporation.

Value	Description
0	Successful execution.
1168	The site collection does not exist.

The **proc_UrlToWebUrl** stored procedure MUST return a single result set as follows.

3.1.5.122.1 Web URL Result Set

The Web URL Result Set returns the store-relative form URL of the Site Collection containing the **@Url** input parameter value. The Web URL Result Set MUST be returned and MUST contain one row.

```
{WebUrl}                        nvarchar(256);
```

{WebUrl}: The store-relative form URL of the Site that contains the **@Url** parameter value. This MUST be an empty string when the **@Url** parameter is invalid or is not contained in the specified site collection, the specified site collection does not exist, or the containing Site is the root site (2) of the site collection. **@Url** is only checked for whether a valid subsite path exists within **@**Url. If **@Url** begins with a '/' the WebURL Result Set returns only the first subsite.

3.1.6 Timer Events

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

3.1.7 Other Local Events

None.

3.2 Web Front-End Client Details

The WFE 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 WFE can maintain the following sets of data for this protocol within object structures. There is no requirement for the data within these structures to be a complete representation of all data maintained on the BEDS, but can be populated as various requests to the BEDS are fulfilled. Data maintained on the WFE can be discarded after individual sequences of requests have completed as part of a response for a higher level event.

- Configuration
- Site Collections
- Sites

Copyright © 2014 Microsoft Corporation.

- Lists
- List Items
- Documents
- Users
- Groups

3.2.2 Timers

A connection timeout timer is set up on the WFE to govern the total connection time for any requests to the BEDS. The amount of time is governed by a timeout value configured on the WFE for all BEDS connections.

3.2.3 Initialization

The WFE MUST validate the user making the request before calling the stored procedure(s). The Site Collection Identifier (section 2.2.1.9) and the User Identifier (section 2.2.1.13) for the user making the request are looked up by the WFE before calling additional stored procedure(s).

3.2.4 Higher-Layer Triggered Events

None.

3.2.5 Message Processing Events and Sequencing Rules

The WFE 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 WFE 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 procedure. SQL queries MUST NOT attempt to add, remove, or update data in any Table or view in the Content or Configuration databases, unless explicitly described in this section.

3.2.6 Timer Events

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

3.2.7 Other Local Events

None.

Copyright © 2014 Microsoft Corporation.

4 Protocol Examples

This section provides specific example scenarios for end-to-end file and permissions management against WSS. These examples describe in detail the process of communication between the various server components involved in the WSS deployment. In conjunction with the detailed protocol documentation described in the reference documents, this information is intended to provide a comprehensive view of how WSS WFE communicate with both Extended UNIX Code (EUC) and back-end database server systems.

4.1 File: GetDocsMetaInfo RPC

This example describes the requests made and responses returned when a user retrieves the Metainfo for a requested site (2), using the Web Service RPC methods provided by the FrontPage Server Extensions: Website Management Specification, as might be generated by a Windows client or the Microsoft SharePoint Designer application.

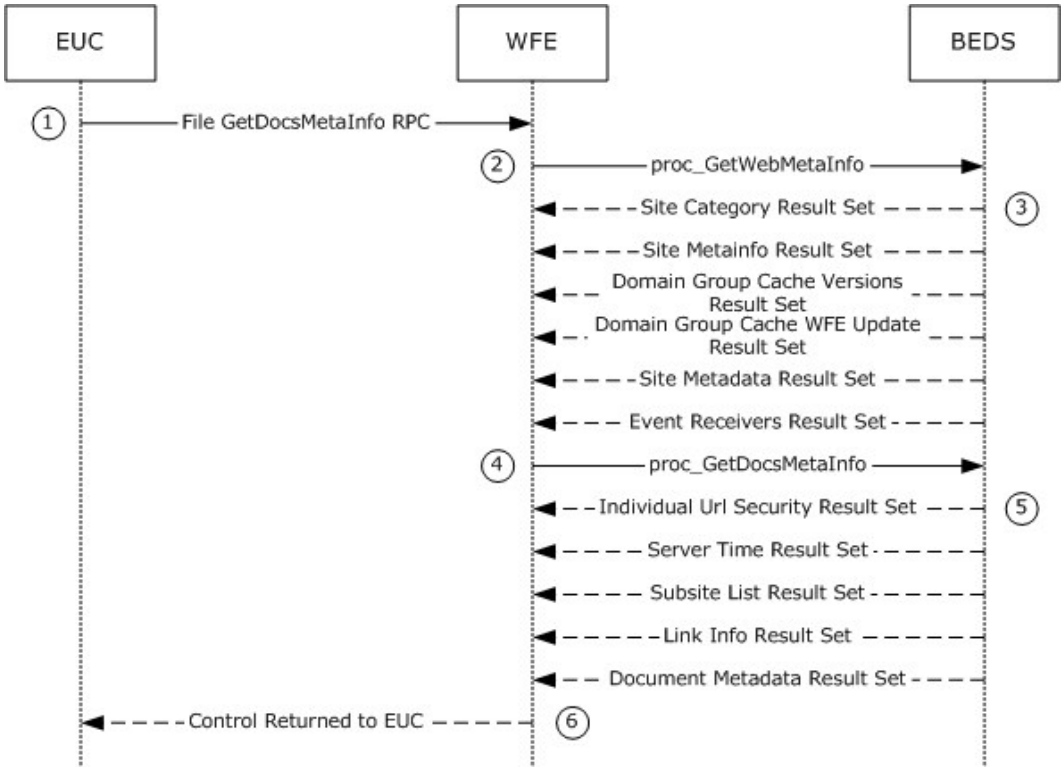


Figure 1: File: GetDocsMetaInfo RPC

This scenario is initiated by a call to the [MC-FPSEWM] command getDocsMetaInfo. For simplicity's sake, this example assumes that a requested document containing some data has already been uploaded to the Shared Documents folder in the root site (2) of the site collection on the WFE. The following actions happen:

1. The [MC-FPSEWM] Remote Procedure Call to proc_getDocsMetaInfo (section 3.1.5.26) is sent to the WFE over HTTP.

2. The WFE in turn requests site metadata information for the requested site (2) from the BEDS. It does this by calling the proc_GetWebMetaInfo (section 3.1.5.40) stored procedure using TDS.

Copyright © 2014 Microsoft Corporation.

3. The BEDS returns six result sets:

   - Site Categories Result Set (section 2.2.5.22), which returns the unordered set of categories defined for content in the requested site (2), one category per row.
   - Site MetaInfo Result Set (section 2.2.5.23), which returns Metainfo for the requested site (2).
   - Domain Group Cache Versions Result Set (section 2.2.5.4), which returns information about the version numbers associated with the external group Map Cache for the site collection containing the requested site (2).
   - Domain Group Cache WFE Update Result Set (section 2.2.5.5), which returns binary data to refresh the external group Map Cache.
   - Site Metadata Result Set (section 2.2.5.23). This returns metadata for the requested site (2).
   - Event Receivers Result Set (section 2.2.5.9). This returns information about the event receivers defined for the requested site (2).

4. The WFE then requests metadata information for the documents contained in the requested site (2). It does this by calling the proc_GetDocsMetaInfo stored procedure using TDS.

5. The BEDS returns five result sets:

   - Individual Url Security Result Set (section 2.2.5.10). This returns information about the security permissions for the documents contained in the requested site.
   - Server Time Result Set (section 2.2.5.18), which returns the current time from the BEDS in UTC.
   - Subsite List Result Set (section 3.1.5.16.1), which returns a list of store-relative form URLs for all subsites of the requested site.
   - Link Info Result Set (section 3.1.5.26.5), which returns a list of all forward links and backward links for the documents contained in the requested site.
   - Document Metadata Result Set (section 2.2.5.6), which returns the metadata for the documents contained in the requested site.

6. A [MC-FPSEWM] HTTP Response is returned to the user, listing information about the requested site.

4.2 File: Open File OM

This example describes the requests and responses made when the WFE opens an existing file stored as a document in a document library on the BEDS with a SharePoint Object Model call.

Copyright © 2014 Microsoft Corporation.

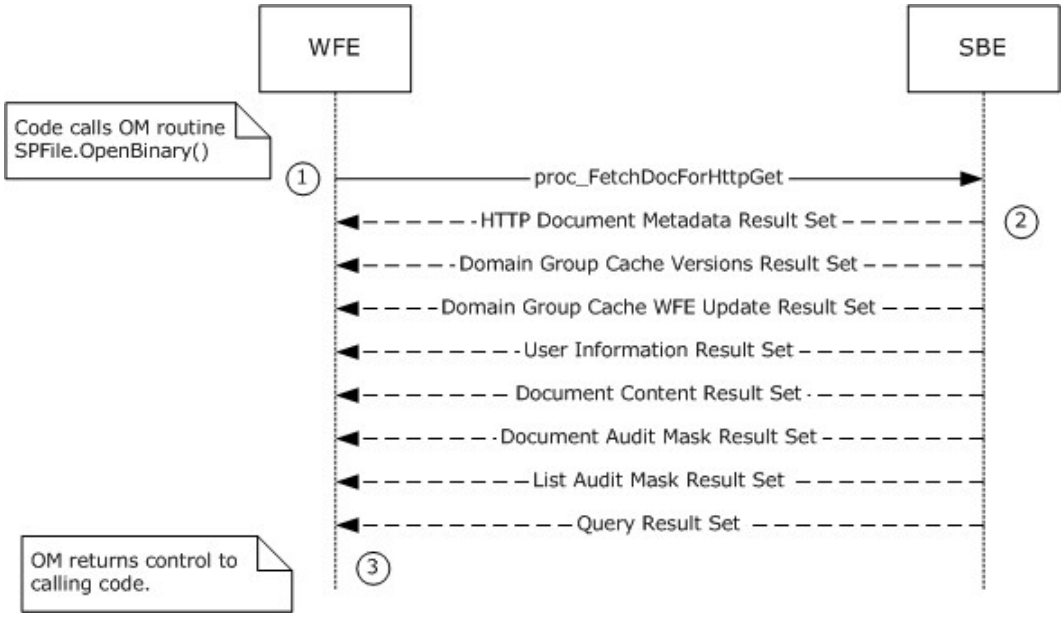


Figure 2: File: Open File OM

This scenario is initiated by a call to the Microsoft.SharePoint.SPFile.OpenBinary() object model command. For simplicity's sake, this example assumes that the file is stored as a document in a document library, and that the requested version is a draft created by the same user who is opening the file. This example assumes that:

- The code has already instantiated the site collection (SPSite), Site (SPWeb), and document library (SPList) objects containing the document to be opened.
- Auditing is disabled for the site collection.
- The current user has File Open permissions for the document.
- Site groups in the site collection do not include any domain groups as members.

The following actions happen:

1. The WFE builds a dynamic query that invokes the proc_FetchDocForHttpGet stored procedure.
2. The BEDS returns a return code of 0, and returns the following result sets:
3. HTTP Document Metadata Result Set (section 3.1.5.15.1). This returns the document metadata needed to further process the document.
4. Domain Group Cache Versions Result Set (section 2.2.5.4). The versions of the Domain Group cache on BEDS and WFE, used to determine if either the BEDS or WFE has more up-to-date information about external group membership in Roles, which are stored as site groups.
5. Domain Group Cache WFE Update Result Set (section 2.2.5.5). Used to update the WFE's external group map cache if needed. Under our assumptions, this result set is empty, since no domain groups are members of any roles in the site collection.
6. User Information Result Set (section 3.1.5.15.5). Used to establish that the current user has permissions to open the file.

Copyright © 2014 Microsoft Corporation.

7. Document Content Stream Result Set (section 3.1.5.15.8). Includes the document stream containing the binary file content for the current version of the document visible to the user, along with additional document metadata.

8. Site Audit Mask Result Set (section 2.2.5.20). Under our assumptions, Auditing is not enabled on the site collection, so the SiteGlobalAuditMask column is NULL.

9. List Audit Mask Result Set (section 3.1.5.17.10), containing auditing information for the document's containing document library. Under our assumptions, auditing is not enabled for the site collection, so the fields containing audit masks are NULL.

10. A Dynamic Query Result Set containing a single row with a single unnamed column, holding the value of the @Level output parameter from the stored procedure proc_FetchDocForHttpGet. The value is 2, indicating that the content returned is from the latest draft version.

11. The OM returns control to the calling program with the array of bytes for the document stream of the requested file.

4.3 Group Add User To Site Group OM

This example describes the requests made when a user is added to a site group using SharePoint Object Model code running on the WFE.

Copyright © 2014 Microsoft Corporation.

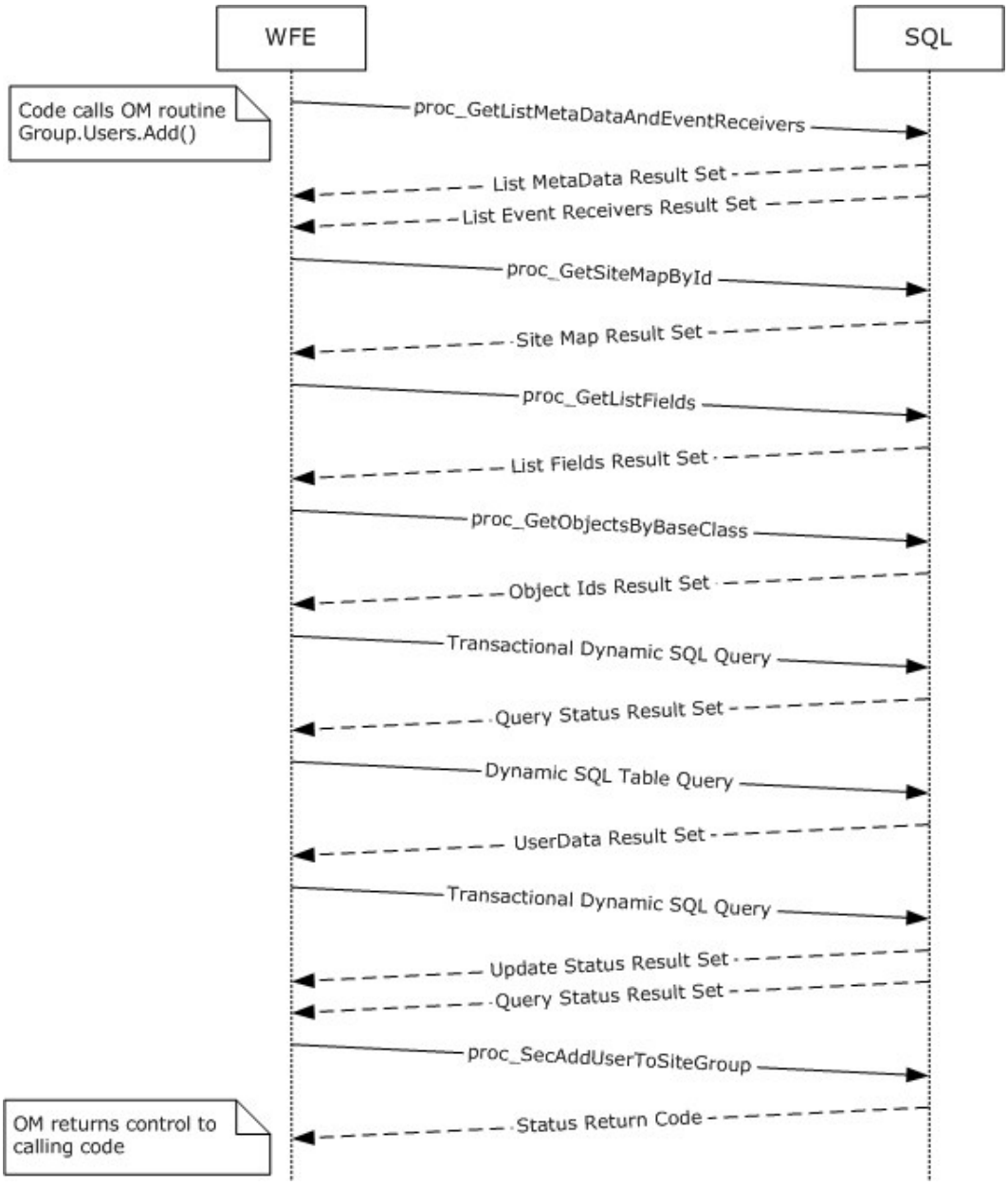


Figure 3: Group Add User To Site Group OM

This scenario is initiated by a call to the object model command SPGroup.Users.Add(). For simplicity's sake, this example assumes that:

- The code has already instantiated the site collection (SPSite), Web (SPWeb), and Group (SPGroup) objects, which contain the site group for this session.
- The instantiated objects have not yet populated information about the user information list.
- The user to be added to the group is currently a user in the site collection.

Copyright © 2014 Microsoft Corporation.

The following actions happen:

1. The WFE first fetches the properties for the "User Information List" of the target site collection, a SharePoint list (1) that contains information about users and groups registered in a site collection. It does this by calling the stored procedure proc_GetListMetaDataAndEventReceivers (section 3.1.5.30) using TDS.

2. The BEDS returns two result sets, which include the List Metadata (section 2.2.5.12) and Event Receivers (section 2.2.5.9) for the specified user information list.

3. The WFE determines the site map for the site collection by calling the configuration database stored procedure proc_getSiteMapById (section 3.1.5.36) using TDS.

4. The BEDS returns a single Site Map By Id Result Set (section 3.1.5.36.1), which includes the site map for the specified site collection.

5. If the WFE determines that the user information list has not been populated in the current SPSite object, it requests the list field information for the site collection's user information list by calling the proc_GetListFields (section 3.1.5.29) stored procedure using TDS.

6. The BEDS returns a single Fields Information Result Set (section 3.1.5.29.1), which includes the field (2) information for the specified user information list.

7. If the WFE determines that it needs to populate the list of SPFeatures, the list of child objects that inherit from the base class SPFeatureDefinition will be populated by calling the configuration database stored procedure proc_getObjectsByBaseClass (section 3.1.5.32) using TDS.

8. The BEDS returns a single Object ID Result Set (section 3.1.5.32.1), which includes a list of child object identifiers for the specified base class and parent object.

9. The WFE builds a transactional dynamic SQL query to add the user to the site collection and add the user's property information to the user information list. This query is sent to the SQL server using TDS. On the SQL server the following actions occur:

   - The query begins a new SQL transaction.
   - The query attempts to add the user to the site collection using the stored procedure proc_SecAddUser (section 3.1.5.47).
   - The query checks if the user exists in the site collection's user information list.
   - If the user is not found in the site collection's user information list, the query attempts to add the user's properties to the site collection's user information list using the stored procedure proc_AddListItem (section 3.1.5.4).
   - The query rolls back the SQL transaction if the previous procedures were not successful, or it commits the transaction if they were successful.

10. The BEDS returns a single result set indicating the status of the actions within the query and the output parameters from the proc_SecAddUser command.

11. The WFE queries the list of users in the user information list for the site collection by building a SQL Batch call to the UserData view (section 2.2.7.8), which is then sent to the SQL server using TDS.

12. The BEDS returns a single result set with a list of items in the UserData View.

Copyright © 2014 Microsoft Corporation.

13. The WFE builds a transactional dynamic SQL query to add or update the list item that represents the user with the user's properties. This query is sent to the SQL server using TDS. On the SQL server the following actions occur:

    - The query begins a new SQL transaction.
    - The query checks if the user exists in the user information list for the site collection.
    - If the user is not found in the user information list for the site collection, then the query attempts to add the user's properties to the site collection's user information list using the stored procedure proc_AddListItem.
    - Otherwise, if the user is found in the site collection's user information list, the query attempts to update the user's properties in the site collection's user information list using the stored procedure proc_UpdateListItem (section 3.1.5.119).
    - The query rolls back the SQL transaction if the previous procedures were not successful, or it commits the transaction if they were successful.

14. The BEDS returns two result sets, which contain the output and return codes from the Add List Item or Update List Item commands.

15. The WFE adds the user to the specified site group by calling the stored procedure proc_SecAddUserToSiteGroup (section 3.1.5.48) using TDS.

16. The BEDS supplies a return code indicating success or failure of the procedure.

4.4 Group Update Site Group Properties OM

This example describes the interactions made when properties are updated for a particular site group.

Copyright © 2014 Microsoft Corporation.

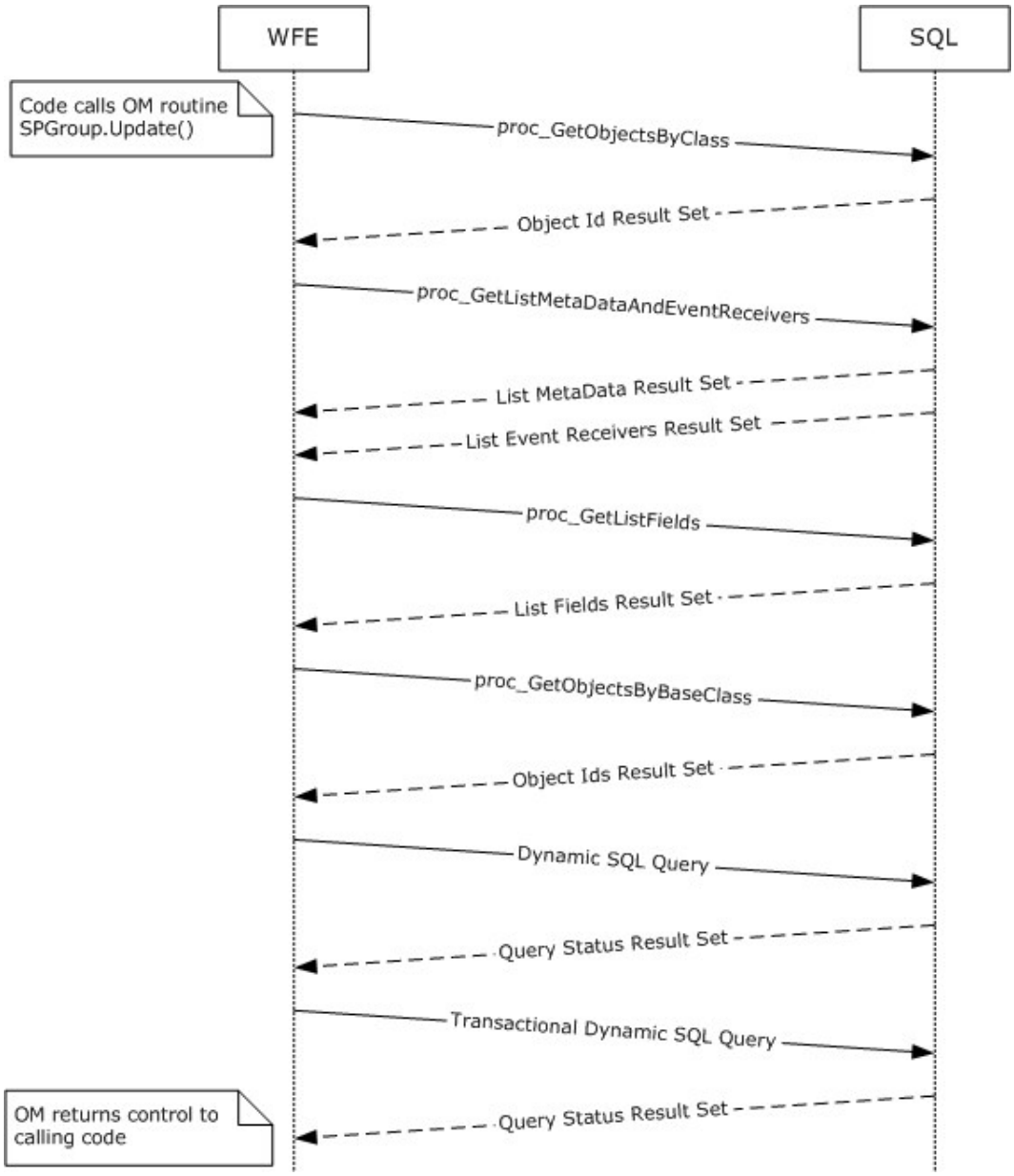


Figure 4: Group Update Site Group Properties OM

This scenario is initiated by a call to the object model command SPGroup.Update(). For simplicity's sake, this example assumes that:

- the code has already instantiated the site collection (SPSite) and site (SPWeb) objects for this session.
- the site group to be updated is in the site collection and is a member of the site.

The following actions happen:

Copyright © 2014 Microsoft Corporation.

1. The WFE determines if the site has the directory management service enabled with a call to the Configuration Database stored procedure proc_GetObjectsByClass (section 3.1.5.33) using TDS.

2. The BEDS returns a single Object ID Result Set (section 3.1.5.33.1) row, which includes a value set if the directory management service is enabled.

3. The WFE fetches the properties for the target site collection's user information list, a SharePoint list (1) containing information about users and groups registered in a site collection. It does this by calling the stored procedure proc_GetListMetaDataAndEventReceivers using TDS.

4. The BEDS returns two result sets, which include the List Metadata (section 2.2.5.12) and Event Receivers (section 2.2.5.9) for the specified user information list.

5. If the WFE determines that the user information list has not been populated in the current SPSite object, it requests the list field (2) information for the site collection's user information list by calling the proc_GetListFields stored procedure using TDS.

6. The BEDS returns a single Field Information Result Set, which includes the field information for the specified user information list.

7. If the WFE determines that it needs to populate the list of SPFeatures, the list of child objects that inherit from the base class SPFeatureDefinition is populated by calling the configuration database stored procedure proc_getObjectsByBaseClass using TDS.

8. The BEDS returns a single object ID result set, which includes a list of child object identifiers for the specified base class and parent object.

9. The WFE builds a dynamic SQL query to select existing information for the site group using TDS.

10. The BEDS returns a single result set, which includes existing data for the user.

11. The WFE builds a transactional dynamic SQL query to update the site group information in the site collection. This query is sent to the SQL server using TDS. On the SQL server the following actions occur:

    - The query begins a new SQL transaction.
    - The query attempts to load the site group information using the stored procedure proc_SecSetSiteGroupProperties (section 3.1.5.107).
    - If the site group is not in the list of site members, the query invokes proc_AddListItem with the updated information.
    - Otherwise, the query attempts to update the site group's properties in the site collection user information list, using the stored procedure proc_UpdateListItem (section 3.1.5.119).
    - The query rolls back the SQL transaction if the previous procedures were not successful, or it commits the transaction if they were successful.

12. The BEDS returns a single rResult set, which indicates the return code status of the actions within the query.

4.5 Security: Add User to Document Library via Object Model

This example describes the requests made when a user is added to the "contributor" role of a document library that has its own scope for independently managed permissions.

Copyright © 2014 Microsoft Corporation.

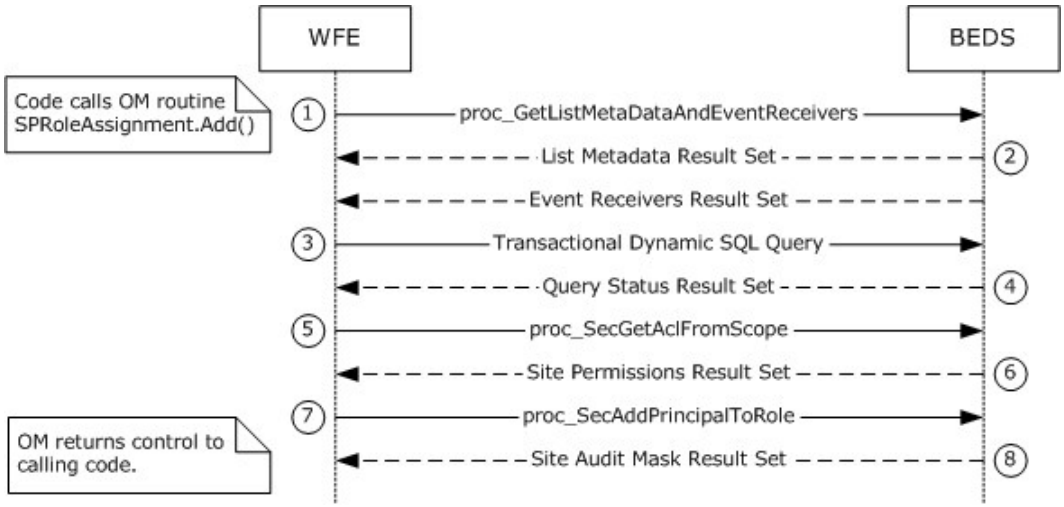


Figure 5: Add User to Document Library via Object Model

This scenario is initiated by a call to the object model command SPRoleAssignmentCollection.Add(SPRoleAssignment).

For simplicity's sake, this example assumes that the code has already instantiated the necessary site collection (SPSite), site (SPWeb), and list (1) (SPList) objects, as well as the role (SPRoleDefinition), role bindings (SPRoleDefinitionBindingCollection) and role assignment (SPRoleAssignment) objects, in order to construct a representation of the role within the document library to which the user will be added.

1. The WFE first fetches the properties for the user information list of the site collection. The user information list is a SharePoint list (1) containing information about users in the site collection. It does this by calling the stored procedure proc_GetListMetaDataAndEventReceivers (section 3.1.5.30) using TDS.

2. The BEDS returns two result sets. The List Metadata Result Set MUST return a single row of data with the metadata for the list. The second result set is the Event Receivers Result Set (section 2.2.5.9). In this example, there are no registered event receivers, and so zero rows are returned.

3. The WFE builds a transactional dynamic SQL query to add the user to the document library's "Contributor" role and to add or update the user's property information in the user information list. This query is sent to the SQL server using TDS. On the SQL server, the following actions occur:

4. The query begins a new SQL transaction.

5. The query attempts to add the user to the site collection's user information list using the stored procedure proc_SecAddUser (section 3.1.5.47).

6. The query checks if the user exists in the site collection's user information list.

7. If the user is not found in the site collection's user information list, then the query attempts to add the user's properties to the site collection's user information list using the stored procedure proc_AddListItem.

8. The query rolls back the SQL transaction if the previous procedures were not successful, or it commits the transaction if they were successful.

Copyright © 2014 Microsoft Corporation.

9. The BEDS returns a single result set indicating the status of the actions within the query and the output parameters from the proc_SecAddUser stored procedure.

10. The WFE fetches the list's scope ACL and anonymous user permission information by calling the stored procedure proc_SecGetAclFromScope (section 3.1.5.58) using TDS.

11. The BEDS returns an ACL and Permissions Result Set (section 3.1.5.17.2), which lists ACL and permission information for the list's scope.

12. The WFE adds the user to the list's "contributor" role assignment membership site group by calling the stored procedure proc_SecAddPrincipalToRole (section 3.1.5.45) using TDS.

13. The BEDS returns a Site Audit Mask Result Set (section 3.1.5.45.1), which lists Auditing Flags information for the site collection and document library.

4.6 Security: Break Web Inheritance OM

This example describes the requests made to create unique security role assignments for a site, rather than inheriting them from a parent.

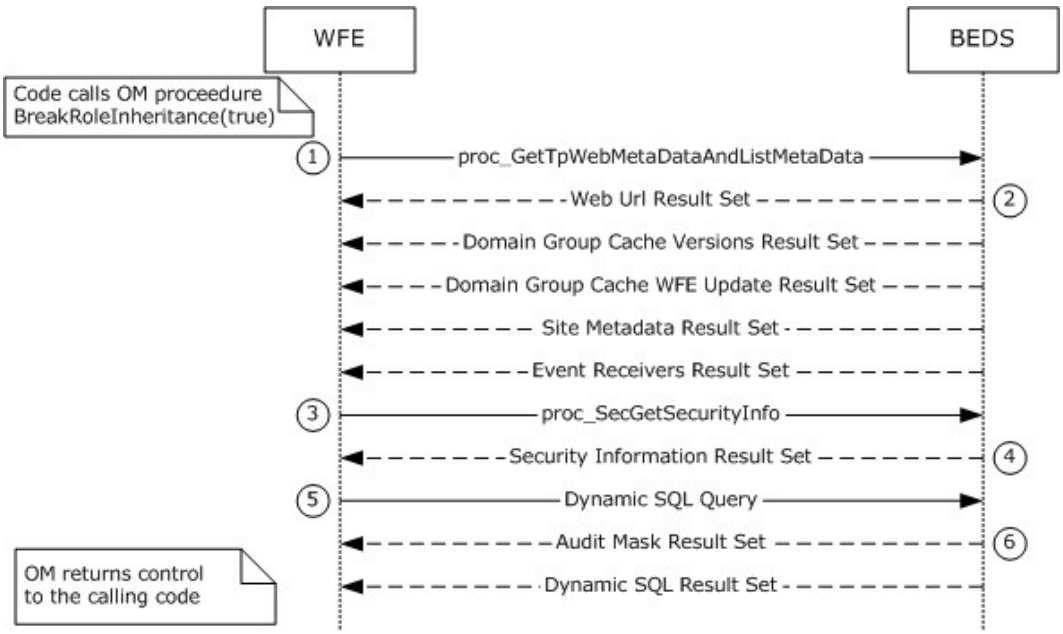


Figure 6: Break Web Inheritance OM

This scenario is initiated by a call to the object model command spweb.BreakRoleInheritance(true). For simplicity's sake, this example assumes that:

- the code has already instantiated the site collection (SPSite), the parent site and child subsite (SPWeb) Objects for this session, and
- the child subsite is initially in the same scope as its parent site.

The following actions happen:

1. The WFE first retrieves metadata for the requested child site. It does this by calling the proc_GetTpWebMetaDataAndListMetaData (section 3.1.5.37) stored procedure using TDS.

Copyright © 2014 Microsoft Corporation.

2. The BEDS returns the following five result sets:

   - Web URL Result Set (section 3.1.5.37.1). This contains the store-relative form URL of the root of the requested child site.
   - Domain Group Cache Versions Result Set (section 2.2.5.4). This contains information about the version numbers associated with the External Group Map Cache for the requested site.
   - Domain Group Cache WFE Update Result Set (section 2.2.5.5). This returns the binary data needed to refresh the external group Map Cache.
   - Site Metadata Result Set (section 2.2.5.23). This contains metadata for the requested site.
   - Event Receivers Result Set (section 2.2.5.9). This contains information about the event receivers defined for the requested site.

3. The WFE then retrieves security permissions information about the requested site. It does this by calling the proc_SecGetSecurityInfo (section 3.1.5.77) stored procedure using TDS.

4. The BEDS returns the Security Information Result Set (section 3.1.5.77.1), which consists of information about security permissions about the requested site.

5. The WFE then builds a Dynamic SQL Query to convert the requested site to use unique permissions (as opposed to inheriting those permission from the Parent site). It does this by calling the proc_SecChangeToUniqueScope (section 3.1.5.52) stored procedure using TDS.

6. The BEDS returns the following two result sets:

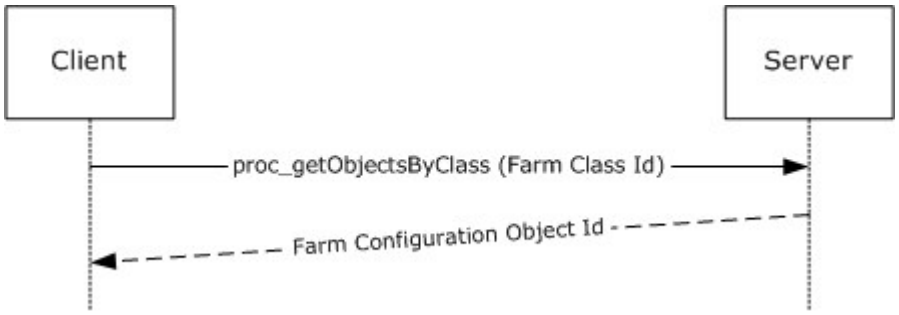
   - SiteAudit Mask Result Set (section 3.1.5.52.1), containing information about the Audit Flags (section 2.2.2.1) set for the requested child site.
   - A Dynamic SQL Result Set, containing the Scope Identifier (section 2.2.1.8) of the new scope generated for the child site.

4.7 Site Collection Lookup

To allow SharePoint's data storage to scale out, site collections can be stored in many content databases. This example illustrates the protocol operations needed to find and connect to a specific content database given a site collection URL. An existing connection to the configuration database using lower-level protocols is assumed.

4.7.1 Retrieving the Farm Id

The example begins by calling proc_GetObjectsByClass (section 3.1.5.33) with the Farm Class ID.



Copyright © 2014 Microsoft Corporation.

Figure 7: Retrieving the Farm ID

This call returns a result set including the Configuration Object ID of the Farm Configuration Object. proc_getObjectsByClass can return result sets with multiple rows, but this implementation of the protocol only ever stores one configuration object (section 2.2.6.1) with the Farm Class ID and ignores all but the first row if multiple rows are returned.

4.7.2 Retrieving the Alternate URL Collection Ids

Next, the Farm Configuration Object Id and the Alternate URL Collection Class Id are passed to proc_getObjectsByBaseClass.

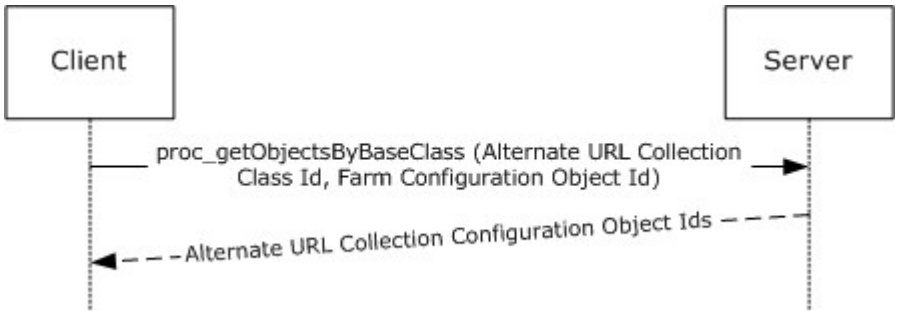


Figure 8: Web Service Lookup Operations

Because the implementation defined the returned Farm Configuration Object to be the Parent of all Alternate URL Collections, this call returns the Configuration Object Ids of all Alternate URL Collections stored in the Configuration Database.

4.7.3 Retrieving the Alternate URL Collections

The retrieved Alternate URL Collection Configuration Object Ids are then passed to proc_getObject to retrieve the full contents of the Alternate URL Collection Configuration Objects.

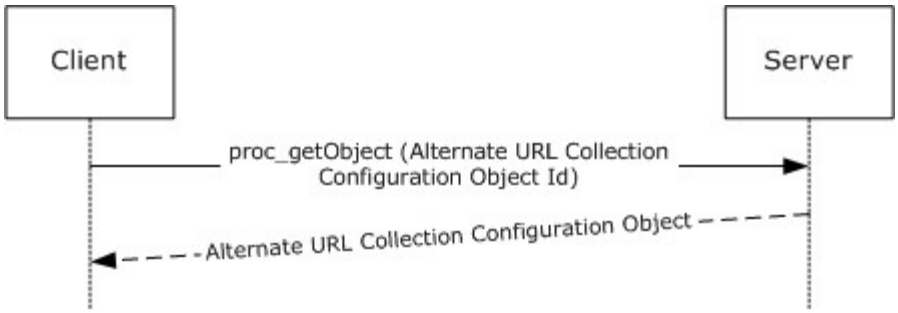


Figure 9: Retrieving the Alternate URL Collections

4.7.4 Alternate URL Matching

At this point, the client can execute the XPath query against the properties of each of the Alternate URL Collections (section 2.2.6.1.7.1) returned to extract all of the alternate URLs. Each of these URLs is then compared against the portion of the incoming URL beginning with the scheme component and ending with the authority component (i.e. "http://example.com:80"). If a match is found the configuration object ID of the Alternate URL Collection containing the matching URL is

Copyright © 2014 Microsoft Corporation.

stored for later use as the request Alternate URL Collections. Otherwise, the site collection lookup operation terminates.

4.7.5 Retrieving the Web Service Ids

Next, the Farm Configuration Object Id returned and the specified Web Service Class Id are passed to proc_getObjectsByBaseClass.

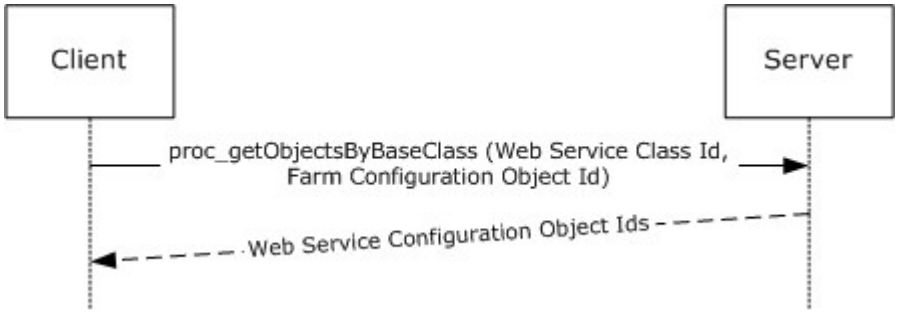


Figure 10: Web Service Lookup Operations

Because the implementation defined the Farm Configuration Object to be the Parent of all Web Services, this call returns the Configuration Object Ids of all Web Services stored in the Configuration Database.

4.7.6 Retrieving the Web Application Ids

In this step, the specified web application Class Id is passed to proc_getObjectsByBaseClass with each of the retrieved Web Service Configuration Object Ids. This returns the Configuration Object Ids of all web applications (1) in the Configuration Database.

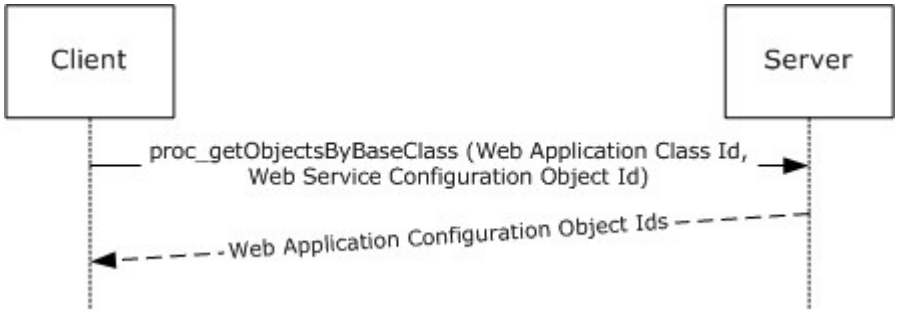


Figure 11: Retrieving the Web Application Ids

4.7.7 Retrieving the Web Applications

The retrieved Web Application Configuration Object Ids are then passed to proc_getObject to retrieve the full contents of the Web Application Configuration Objects.

Copyright © 2014 Microsoft Corporation.

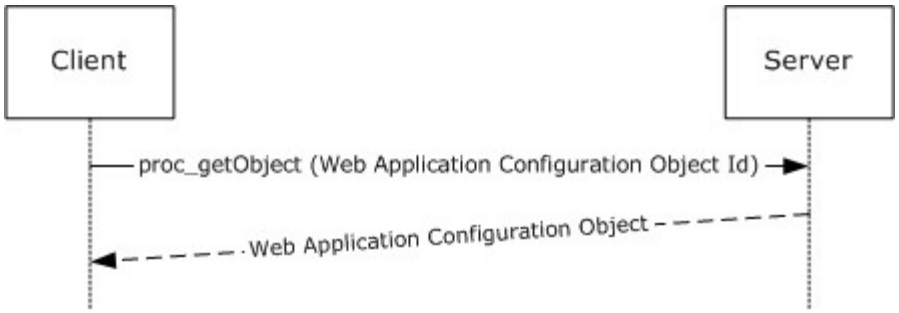


Figure 12: Retrieving the Web Applications

4.7.8 Web Application Lookup

At this point, the client can execute the Web Application Alternate URL Collection XPath Query to extract the Alternate URL Collection (section 2.2.6.1.7.1) Ids associated with each retrieved web application (1). These Alternate URL Collection IDs are compared against the Request Alternate URL Collection ID. If a match is found, the associated web application (1) is used as the Request Web Application for the remainder of site collection Lookup. Otherwise, site collection lookup terminates.

4.7.9 Prefix Matching

Web applications (1) contain a set of site collection Prefixes which contain a name and a type. site collection Lookup extracts these values from the Request Web Application Properties using XPath Queries.

The Prefix names are URL Path Components used to determine which portion of the incoming URL Path Component is the server-relative URL of the site collection. This is done by matching all of the Prefixes in the Request Web Application against the start of the Path Component of the incoming URL. If more than one Prefix matches the beginning of the incoming URL Path Component, the longest matching Prefix is used.

There are two types of Prefix: wildcard and explicit. A web application (1) can contain any combination of both types.

4.7.9.1 Explicit Prefixes

An explicit prefix indicates that the portion of the Path Component up to and including the Prefix are included in the site collection server-relative URL. For example, if a user requests http://example.com/sitename/web/list/document.htm and if the web application (1) corresponding to http://example.com contains an explicit prefix named "sitename", then "/sitename" is the server-relative URL of the site collection.

Incoming URL	Web Application Explicit Prefixes	Resulting Site Collection Server-Relative URL
http://example.com/a/b/c.htm	"a"	"/a"
http://example.com/a/b/c.htm	"a", "a/b"	"/a/b"
http://example.com/a/b.htm	"a", "a/b"	"/a"
http://example.com/a/b.htm	"c"	<No Match>

Copyright © 2014 Microsoft Corporation.

Incoming URL	Web Application Explicit Prefixes	Resulting Site Collection Server-Relative URL
http://example.com/a/b.htm	""	"/"

4.7.9.2 Wildcard Prefixes

A wildcard prefix indicates that the portion of the Path Component up to and including the first Path Component segment following the Prefix are included in the site collection name. For example, if a user makes a request for http://example.com/sites/sitename/web/list/document.htm, and if the web application (1) corresponding to http://example.com contains a Wildcard Prefix named "sites", then "/sites/sitename" is the server-relative URL of the site collection.

Incoming URL	Web Application Wildcard Prefixes	Resulting Site Collection Server-Relative URL
http://example.com/a/b/c/d.htm	"a", "a/b	"/a/b/c"
http://example.com/a/b.htm	"a", "a/b"	<No Match>
http://example.com/a/b.htm	""	"/a"

4.7.10 Site Collection Id Lookup

Once the site collection URL is determined, it is passed to proc_getSiteMap (section 3.1.5.35), along with the request web application ID.

Figure 13: Site Collection Id Lookup

A site collection ID is returned along with the configuration object ID of the content database in which the site collection content is stored. If the specified combination site collection URL and web application ID cannot be found in the configuration database, the site collection does not exist and site collection lookup terminates.

4.7.11 Building Content Database Connection String

At this point, the only step that remains is to establish a connection to the content database. This requires a content database connection string, which is built by combining the following components:

Copyright © 2014 Microsoft Corporation.

4.7.11.1 Name

Once the content database ID is known, it is passed to proc_GetObject, which returns the content database configuration object (section 2.2.6.1).

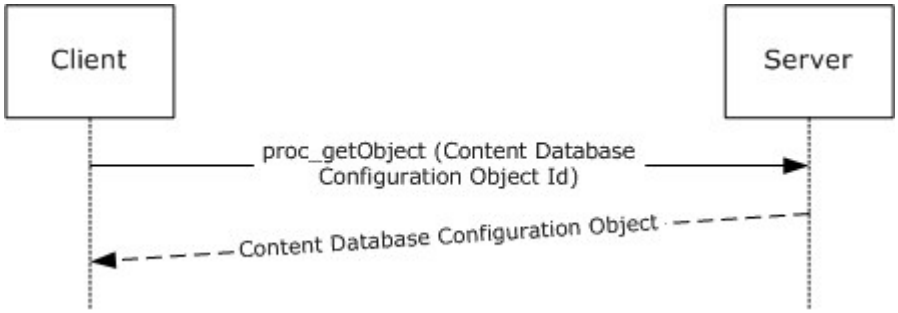


Figure 14: Name

The name field (2) of the content database configuration object is used in the connection string as the content database name.

4.7.11.2 Credentials

The optional Username and Password are extracted from the Properties of the Content Database configuration object (section 2.2.6.1) using XPath queries.

4.7.11.3 Instance

The ParentId of the content database configuration object (section 2.2.6.1) is passed in another call to proc_GetObject, which returns the database service instance configuration object, the name field (2) of which is used as the Database Server Instance.

Figure 15: Database Server Instance

4.7.11.4 Server Address

The ParentId of the Database Service Instance Configuration Object is then passed to the one final call to proc_getObject, which a server configuration object (section 2.2.6.1). The name field (2) of the server configuration object is used as the Server Address component of the connection string.

Copyright © 2014 Microsoft Corporation.

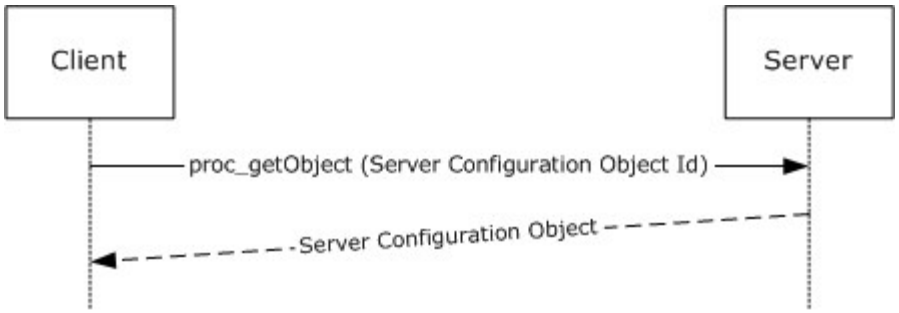


Figure 16: Server Address

At the end of this step, the incoming URL has been successfully translated into a site collection URL, a site collection Id, and a Content Database Connection String. These components are then ready to be used to call other stored procedures in this and other protocols.

4.8 User Update User Properties OM

This example describes the interactions made when site member properties are updated for a particular user.

Copyright © 2014 Microsoft Corporation.

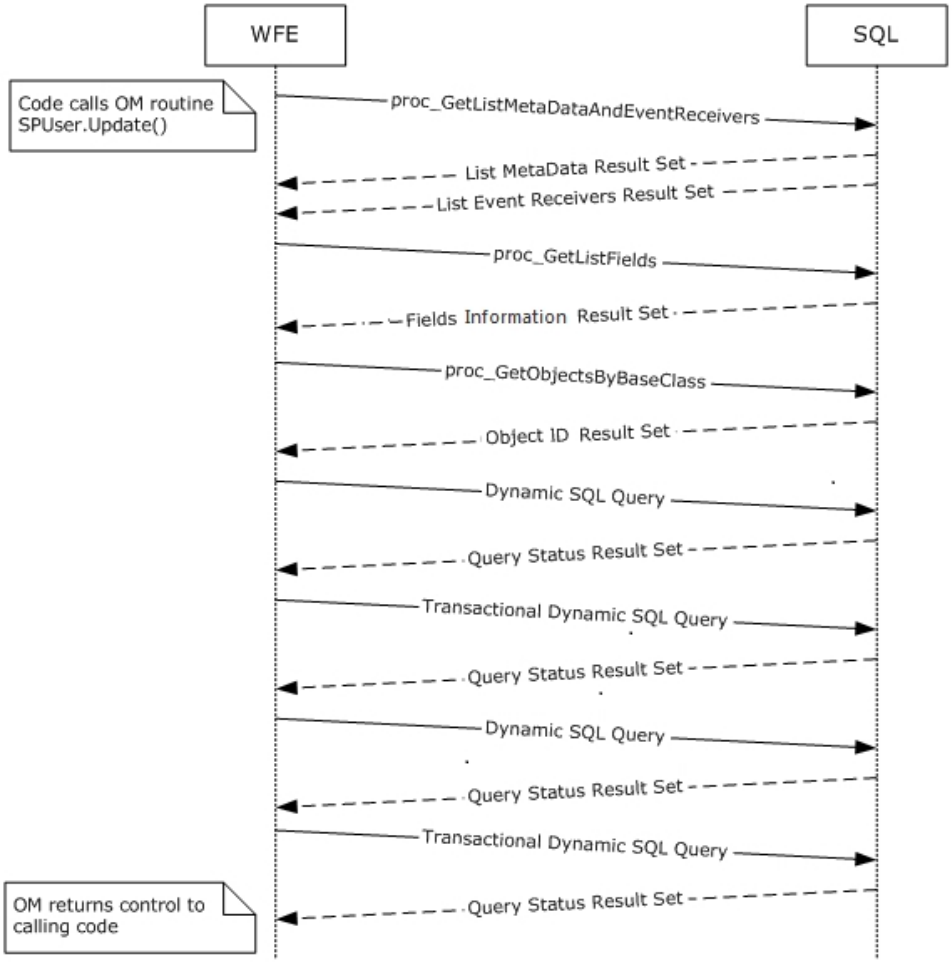


Figure 17: User Update User Properties OM

This scenario is initiated by a call to the object model command SPUser.Update(). For simplicity's sake, this example assumes that:

- The code has already instantiated the site collection (SPSite) and Web (SPWeb) Objects for this session, and
- The user to be updated is in the site collection and a member of the site (2).

The following actions happen:

1. The WFE fetches the properties for the target site collection's user information list, a SharePoint list (1) containing information about users and groups registered in a site collection. It does this by calling the stored procedure proc_GetListMetaDataAndEventReceivers (section 3.1.5.30) using TDS.
2. The BEDS returns two result sets, which include the list metadata result set (section 3.1.5.30.1) and the event receivers result set (section 3.1.5.30.4) for the specified user information list.

Copyright © 2014 Microsoft Corporation.

3. If the WFE determines that the user information list has not been populated in the current SPSite object, it requests the list field information for the site collection's user information list by calling the stored procedure proc_GetListFields using TDS.

4. The BEDS returns a single fields information result set, which includes the field information for the specified user information list.

5. If the WFE determines that it needs to populate the list of SPFeatures, the list of child objects that inherit from the base class SPFeatureDefinition is populated by calling the configuration database stored procedure proc_getObjectsByBaseClass using TDS.

6. The BEDS returns a single Object ID result set, which includes a list of child object identifiers for the specified base class and parent object.

7. The WFE builds a dynamic SQL query to select existing information for the user using TDS.

8. The BEDS returns a single result set, which includes existing data for the user.

9. The WFE builds a transactional dynamic SQL query to update the user information in the site (2). This query is sent to the SQL server using TDS. On the SQL server the following actions occur:

   - The query begins a new SQL transaction.
   - The query attempts to update the user information using the stored procedure proc_SecUpdateUser (section 3.1.5.113).
   - The query attempts to update the user's properties in the site's user information list using the stored procedure proc_UpdateListItem (section 3.1.5.119).
   - The query rolls back the SQL transaction if the previous procedures were not successful, or it commits the transaction if they were successful.

10. The BEDS returns a single result set, which indicates the return code status of the actions within the query.

11. The WFE builds a dynamic SQL query to select updated information for the user using TDS.

12. The BEDS returns a single result set, which includes the data for the user.

13. The WFE builds a transactional dynamic SQL query to update the user information in the site. This query is sent to the SQL server using TDS. On the SQL server the following actions occur:

    - The query begins a new SQL transaction.
    - The query attempts to update the site collection's user list data using the stored procedure proc_UpdateListItem.
    - Then the query attempts to update the user's properties in the site collection's user information table using the stored procedure proc_UpdateUserInfoInTableFromRowUpdater (section 3.1.5.121).
    - The query rolls back the SQL transaction if the previous procedures were not successful, or it commits the transaction if they were successful.

14. The BEDS returns a successful return code status.

Copyright © 2014 Microsoft Corporation.

4.9 Version Negotiation

The following scenario is an example of the protocol version negotiation sequence between a WFE and a content database where the content database version does not match that expected by the WFE.

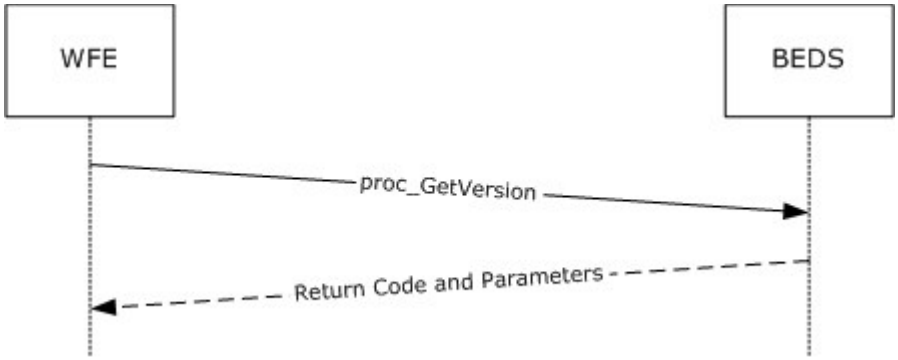


Figure 18: Version Negotiation

1. The WFE calls the proc_GetVersion (section 3.1.5.39) stored procedure in the content database on the BEDS, with the @VersionId parameter set to '6333368D-85F0-4EF5-8241-5252B12B2E50'.

2. The proc_GetVersion stored procedure returns the output parameter @Version with a value of '3.0.149.0' (WSS RTM). The application has a pre-defined version of '3.1.4.0' (WSS SP1), which reflects the current state of the software. Any new database created by this application will have this version. The application determines that the version from the database is smaller (meaning it is older) than the version the WFE expects, and does not perform any further operations against the database.

Copyright © 2014 Microsoft Corporation.

5 Security

5.1 Security Considerations for Implementers

Security for this protocol is controlled by the access rights to the databases on the BEDS, which is negotiated as part of the Tabular Data Stream [MS-TDS] protocol. In a trusted subsystem model, the process running on the WFE uses its own security principal (2) identity to access the content database on the BEDS on behalf of the user, rather than using the account of the user accessing the WFE as a database access account to access the content database. The database access account used by the WFE server is required to have access to the appropriate content database on the BEDS. If the account does not have the correct access rights, access will be denied when attempting to set up the [MS-TDS] connection to the content database, or when calling the stored procedures.

5.2 Index of Security Parameters

The following table includes the security parameters.

Security Parameter	Section
proc_SecAddPrincipalToRole	3.1.5.45
proc_SecAddRoleDef	3.1.5.46
proc_SecAddUser	3.1.5.47
proc_SecAddUserToSiteGroup	3.1.5.48
proc_SecAddWebMembership	3.1.5.49
proc_SecChangeToInheritedList	3.1.5.50
proc_SecChangeToInheritedWeb	3.1.5.51
proc_SecChangeToUniqueScope	3.1.5.52
proc_SecCheckDeletedAccounts	3.1.5.53
proc_SecCloneRoleDefinitions	3.1.5.54
proc_SecCreateSiteGroup	3.1.5.55
proc_SecDecCurrentUsersCount	3.1.5.56
proc_SecGetAccountStatus	3.1.5.57
proc_SecGetAclFromScope	3.1.5.58
proc_SecGetAllAclsForSite	3.1.5.59
proc_SecGetAllGroupsAndMembershipInfo	3.1.5.60
proc_SecGetCompleteWebRoleMemberList	3.1.5.61
proc_SecGetCurrentUsersCount	3.1.5.62
proc_SecGetDomainGroupMapData	3.1.5.63
proc_SecGetGroupById	3.1.5.64

Copyright © 2014 Microsoft Corporation.

Security Parameter	Section
proc_SecGetGroupOwner	3.1.5.65
proc_SecGetGroupSecurityScopes	3.1.5.66
proc_SecGetIndividualUrlSecurityCheckEventReceivers	3.1.5.67
proc_SecGetPrincipalByEmail	3.1.5.68
proc_SecGetPrincipalById	3.1.5.69
proc_SecGetPrincipalByLogin	3.1.5.70
proc_SecGetPrincipalByLogin20	3.1.5.71
proc_SecGetPrincipalDisplayInformation20	3.1.5.72
proc_SecGetRoleAssignments	3.1.5.73
proc_SecGetRoleBindingsForAllPrincipals	3.1.5.74
proc_SecGetRoleDefs	3.1.5.75
proc_SecGetRoleDefsForPrincipal	3.1.5.76
proc_SecGetSecurityInfo	3.1.5.77
proc_SecGetSiteAdmins	3.1.5.78
proc_SecGetSiteGroupById	3.1.5.79
proc_SecGetSiteGroupByTitle	3.1.5.80
proc_SecGetSiteGroupByTitle20	3.1.5.81
proc_SecGetUserAccountDirectoryPath	3.1.5.82
proc_SecGetUserPermissionOnGroup	3.1.5.83
proc_SecListAllSiteMembers	3.1.5.84
proc_SecListAllWebMembers	3.1.5.85
proc_SecListGroupsInRole	3.1.5.86
proc_SecListScopeGroups	3.1.5.87
proc_SecListScopeUsers	3.1.5.88
proc_SecListSiteGroupMembership	3.1.5.89
proc_SecListSiteGroups	3.1.5.90
proc_SecListSiteGroupsContainingUser	3.1.5.91
proc_SecListSiteGroupsWhichUserOwns	3.1.5.92
proc_SecListUsersInRole	3.1.5.93
proc_SecMigrateUser	3.1.5.94

Copyright © 2014 Microsoft Corporation.

Security Parameter	Section
proc_SecReCalculateWebFGP	3.1.5.95
proc_SecRefreshToken	3.1.5.96
proc_SecRemoveGroup	3.1.5.97
proc_SecRemovePrincipalFromScope	3.1.5.98
proc_SecRemoveRoleDef	3.1.5.99
proc_SecRemoveUserFromScopeByLogin	3.1.5.100
proc_SecRemoveUserFromSite	3.1.5.101
proc_SecRemoveUserFromSiteGroup	3.1.5.102
proc_SecRemoveUserFromSiteGroupByLogin	3.1.5.103
proc_SecResetItemPerm	3.1.5.104
proc_SecResetWebToDefaultRoleDefinition	3.1.5.105
proc_SecResolvePrincipal	3.1.5.106
proc_SecSetSiteGroupProperties	3.1.5.107
proc_SecSetUserAccountDirectoryPath	3.1.5.108
proc_SecSetWebRequestAccess	3.1.5.109
proc_SecUpdateAnonymousPermMask	3.1.5.110
proc_SecUpdateDomainGroupMapData	3.1.5.111
proc_SecUpdateRoleDef	3.1.5.112
proc_SecUpdateUser	3.1.5.113
proc_SecUpdateUserActiveStatus	3.1.5.114

Copyright © 2014 Microsoft Corporation.

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:

- Windows Server 2003 operating system
- Windows Server 2008 operating system
- Windows Server 2008 R2 operating system
- Windows SharePoint Services 3.0

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

<1> Section 2.2.1.7: By default, Windows SharePoint Services 3.0 creates five implementation-specific Role Definitions with associated default Role Identifiers.

Role identifier value	Role definition
1073741825	Guest
1073741826	Reader
1073741827	Contributor
1073741828	Web Designer
1073741829	Administrator

<2> Section 2.2.1.10: By default, Windows SharePoint Services 3.0 creates three implementation-specific site groups with default site group identifiers for use in **role assignments** when provisioning a new site collection: "Site Owners" (3), "Site Visitors" (4), and "Site Members" (5).

<3> Section 2.2.2.8: This flag is not used in Windows SharePoint Services 3.0.

<4> Section 2.2.8.3.3.2: Windows SharePoint Services 3.0 allows a filtering user interface for fields with the aggregation attribute set to merge, but only only in list views where Filter is equal to 1 occurs in the view's query parameter.

<5> Section 2.2.8.3.3.2: Windows SharePoint Services 3.0 can include this attribute with a value of 0.

<6> Section 2.2.8.3.3.2: Windows SharePoint Services 3.0 sets the value MinusSign for some fields.

<7> Section 2.2.8.3.3.3: Windows SharePoint Services 3.0 emits this data if present in the underlying template definition, even if the field type is not of the specified type.

Copyright © 2014 Microsoft Corporation.

<8> Section 2.2.8.3.3.3: Windows SharePoint Services 3.0 sends this data if present in the underlying template definition, even if the field type is not of the specified type.

<9> Section 2.2.8.3.3.3: Windows SharePoint Services 3.0 sends this data if present in the underlying template definition, even if the field type is not of the specified type.

<10> Section 2.2.8.3.3.3: Windows SharePoint Services 3.0 sends this data if present in the underlying template definition, even if the field type is not of the specified type.

<11> Section 2.2.8.3.3.3: Windows SharePoint Services 3.0 sends this data if present in the underlying template definition, even if the field type is not of the specified type.

<12> Section 3.1.5.39: SharePoint Products and Technologies uses text based SQL queries to get the version number which are equivalent in functionality to **proc_GetVersion**.

Copyright © 2014 Microsoft Corporation.

7 Change Tracking

This section identifies changes that were made to the [MS-WSSFO] protocol document between the July 2014 and October 2014 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.
- The removal of a document from the documentation set.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **Editorial** means that the formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the technical content of the document is identical to the last released version.

Major and minor changes can be described further using the following change types:

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.
- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.

Copyright © 2014 Microsoft Corporation.

- Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated.**

Some important terms used in the change type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

The changes made to this document are listed in the following table. For more information, please contact dochelp@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
2.2.2.5 List Flags	Added a definition number for the term "form".	N	Content updated.
2.2.2.8 Rename Flags	Added a definition number for the term "form".	N	Content updated.
2.2.2.13 WSS Rights Mask	Added a definition number for the term "form".	N	Content updated.
2.2.3.14 Page Type	Added a definition number for the term "form".	N	Content updated.
2.2.8.3.3.2 Attributes	Added a definition number for the term "form".	N	Content updated.
	Added a definition number for the term "workflow" where it appears throughout the document.	N	Content updated.

Copyright © 2014 Microsoft Corporation.

8 Index

A

Abstract data model
  client 351
  server 135
Applicability 20

C

Capability negotiation 20
Change tracking 379
Client
  abstract data model 351
  higher-layer triggered events 352
  initialization 352
  message processing (section 3.2.5 352, section 3.2.5 352)
  other local events 352
  overview 351
  sequencing rules (section 3.2.5 352, section 3.2.5 352)
  timer events 352
  timers 352

D

Data model - abstract
  client 351
  server 135

E

Examples
  File - GetDocsMetaInfo RPC 353
  File - Open File OM 354
  Group Add User To Site Group OM 356
  Group Update Site Group Properties OM 359
  overview 353
  Security - Add User to Document Library via Object Model 361
  Security – Break Web Inheritance OM 363
  Site Collection Lookup 364
  User Update User Properties OM 370
  Version Negotiation 373

F

Fields - vendor-extensible 20
File - GetDocsMetaInfo RPC example 353
File - Open File OM example 354

G

Glossary 16
Group Add User To Site Group OM example 356
Group Update Site Group Properties OM example 359

H

Higher-layer triggered events
  client 352
  server 136

I

Implementer - security considerations 374
Index of security parameters 374
Informative references 19
Initialization
  client 352
  server 136
Introduction 16

M

Message processing
  client (section 3.2.5 352, section 3.2.5 352)
  server (section 3.1.5 136, section 3.1.5 136)
Messages
  Result Sets 55
  Simple Data Types 21
  syntax 21
  transport 21

N

Normative references 18

O

Other local events
  client 352
  server 351
Overview
  file operations 19
  user and group operations 19
Overview (synopsis) 19

P

Parameters - security index 374
Preconditions 20
Prerequisites 20
Product behavior 377

R

References 18
  informative 19
  normative 18
Relationship to other protocols 19
Result Sets message 55

S

Copyright © 2014 Microsoft Corporation.

Security
implementer considerations 374
parameter index 374
Security - Add User to Document Library via Object Model example 361
Security - Break Web Inheritance OM example 363
Security – User Update User Properties OM example 370
Sequencing rules
client (section 3.2.5 352, section 3.2.5 352)
server (section 3.1.5 136, section 3.1.5 136)
Server
abstract data model 135
higher-layer triggered events 136
initialization 136
message processing (section 3.1.5 136, section 3.1.5 136)
other local events 351
sequencing rules (section 3.1.5 136, section 3.1.5 136)
timer events 351
timers 136
Simple Data Types message 21
Site Collection Lookup example 364
alternate URL matching 365
building the database connection string 368
prefix matching 367
retrieving the alternate URL collection Ids 365
retrieving the alternate URL collections 365
retrieving the application Ids 366
retrieving the farm Id 364
retrieving the web applications 366
retrieving the Web service Ids 366
site collection Id lookup 368
web application lookup 367
Standards assignments 20
Syntax 21

T

Timer events
client 352
server 351
Timers
client 352
server 136
Tracking changes 379
Transport 21
Triggered events - higher-layer
client 352
server 136

V

Vendor-extensible fields 20
Version Negotiation example 373
Versioning 20

Copyright © 2014 Microsoft Corporation.