

[MS-WSSDM]: Windows SharePoint Services: Content Database Data Migration Communications Protocol Specification

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
04/04/2008	0.1		Initial Availability
06/27/2008	1.0	Major	Revised and edited the technical content
10/06/2008	1.01	Editorial	Revised and edited the technical content
12/12/2008	1.02	Editorial	Revised and edited the technical content
03/18/2009	1.03	Editorial	Revised and edited the technical content
07/13/2009	1.04	Major	Changes made for template compliance
08/28/2009	1.05	Editorial	Revised and edited the technical content
11/06/2009	1.06	Editorial	Revised and edited the technical content
02/19/2010	2.0	Editorial	Revised and edited the technical content
03/31/2010	2.01	Editorial	Revised and edited the technical content
04/30/2010	2.02	Editorial	Revised and edited the technical content
06/07/2010	2.03	Editorial	Revised and edited the technical content
06/29/2010	2.04	Editorial	Changed language and formatting in the technical content.
07/23/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
09/27/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
06/10/2011	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
01/20/2012	2.5	Minor	Clarified the meaning of the technical content.
04/11/2012	2.5	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	2.5	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction	8
1.1 Glossary	8
1.2 References	10
1.2.1 Normative References	10
1.2.2 Informative References	11
1.3 Protocol Overview (Synopsis)	11
1.4 Relationship to Other Protocols	11
1.5 Prerequisites/Preconditions	11
1.6 Applicability Statement	12
1.7 Versioning and Capability Negotiation	12
1.8 Vendor-Extensible Fields	12
1.9 Standards Assignments	12
2 Messages	13
2.1 Transport	13
2.2 Common Data Types	13
2.2.1 Simple Data Types and Enumerations	13
2.2.2 Simple Data Types	13
2.2.2.1 Web Part Moderation	13
2.2.2.2 Web Part Scope	13
2.2.3 Bit Fields and Flag Structures	13
2.2.3.1 Security Group Membership Flag	13
2.2.4 Enumerations	14
2.2.4.1 Deployment Object Type	14
2.2.4.2 Role Type	15
2.2.5 Binary Structures	15
2.2.6 Result Sets	15
2.2.7 Tables and Views	15
2.2.7.1 WebParts Table	15
2.2.7.2 WebPartLists Table	16
2.2.7.3 #WebParts Table	16
2.2.8 XML Structures	16
2.2.8.1 Namespaces	16
2.2.8.2 Simple Types	16
2.2.8.3 Complex Types	16
2.2.8.4 Elements	16
2.2.8.5 Attributes	17
2.2.8.6 Groups	17
2.2.8.7 Attribute Groups	17
3 Protocol Details	18
3.1 Server Details	18
3.1.1 Abstract Data Model	18
3.1.2 Timers	18
3.1.3 Initialization	18
3.1.4 Message Processing Events and Sequencing Rules	18
3.1.4.1 proc_DeplAddExportObject	18
3.1.4.2 proc_DeplAddExportObjectDependencies	20
3.1.4.3 proc_DeplAddExportObjectLinks	21
3.1.4.3.1 Link Count Result Set	21

3.1.4.4	proc_DeplAddIncrementalSearchScope.....	22
3.1.4.5	proc_DeplAddSite	22
3.1.4.6	proc_DeplCalculateChildrenToExport	23
3.1.4.7	proc_DeplCalculateIncrementalSearchScope	24
3.1.4.8	proc_DeplExportGroups	24
3.1.4.8.1	Security Group Information Result Set	25
3.1.4.8.2	Security Group Memberships Result Set	25
3.1.4.9	proc_DeplExportGroupX	26
3.1.4.9.1	Security Group Change Result Set	26
3.1.4.10	proc_DeplExportRoleAssignments	27
3.1.4.10.1	Scope Information Result Set.....	27
3.1.4.10.2	Role Assignment Information Result Set.....	28
3.1.4.11	proc_DeplExportRoleAssignmentX	29
3.1.4.11.1	Role Assignment Change Result Set.....	29
3.1.4.12	proc_DeplExportRoles	30
3.1.4.12.1	Role Information Result Set	31
3.1.4.13	proc_DeplExportRoleX	31
3.1.4.13.1	Role Change Result Set	32
3.1.4.14	proc_DeplExportUsers	33
3.1.4.14.1	Security Principal Information Result Set	33
3.1.4.15	proc_DeplExportUserX.....	34
3.1.4.15.1	Security Principal Change Result Set	35
3.1.4.16	proc_DeplFindScopeObject.....	35
3.1.4.17	proc_DeplFindScopeObjectById	36
3.1.4.18	proc_DeplFixWebPartsOnListCreation	36
3.1.4.19	proc_DeplGetContentTypeData	37
3.1.4.19.1	Content Type Information Result Set	37
3.1.4.20	proc_DeplGetContentTypesToParse	37
3.1.4.20.1	Content Type List Result Set	38
3.1.4.21	proc_DeplGetDocLibRootFolderUrl	38
3.1.4.22	proc_DeplGetFileData	39
3.1.4.22.1	File Information Result Set	39
3.1.4.22.2	Latest File Versions Result Set	41
3.1.4.22.3	Previous File Versions Result Set	43
3.1.4.22.4	File Web Part Result Set	43
3.1.4.22.5	Web Part Personalization Result Set.....	46
3.1.4.22.6	File Links Result Set.....	46
3.1.4.22.7	File Event Receivers Result Set	47
3.1.4.23	proc_DeplGetFolderById	48
3.1.4.23.1	Folder Location Result Set	49
3.1.4.24	proc_DeplGetFolderData	49
3.1.4.24.1	Folder Information Result Set	50
3.1.4.25	proc_DeplGetIncrementalChanges	51
3.1.4.26	proc_DeplGetListData	52
3.1.4.26.1	List Information Result Set	53
3.1.4.26.2	List Web Parts Result Set	56
3.1.4.26.3	Event Receivers Result Set	58
3.1.4.26.4	Deleted Fields Result Set	59
3.1.4.26.5	Null List Result Set	60
3.1.4.27	proc_DeplGetListItemAndReserveId	60
3.1.4.28	proc_DeplGetListItemByDocId	61
3.1.4.29	proc_DeplGetListItemByUrl	62
3.1.4.30	proc_DeplGetListItemData	62

3.1.4.30.1	List Item Data Result Set	63
3.1.4.30.2	Field Data Result Set	65
3.1.4.30.3	Multi Lookup Result Set	66
3.1.4.30.4	List Item Versions Result Set	67
3.1.4.30.5	Versions Fields Data Result Set	69
3.1.4.30.6	Versions Multi Lookup Result Set	70
3.1.4.30.7	List Item Links Result Set	71
3.1.4.30.8	Attachments Result Set	71
3.1.4.30.9	Missing List Result Set	72
3.1.4.30.10	List Item Event Receivers Result Set	73
3.1.4.31	proc_DeplGetLookupItemData	74
3.1.4.31.1	Lookup Data Result Set	74
3.1.4.32	proc_DeplGetLookupListUrl	75
3.1.4.33	proc_DeplGetNextObjectBatch	75
3.1.4.33.1	Export Object Batch Result Set	75
3.1.4.34	proc_DeplGetParentWebIdForList	76
3.1.4.34.1	Parent Site Result Set	77
3.1.4.35	proc_DeplGetParentWebIdFromDocId	77
3.1.4.35.1	Document Container Result Set	77
3.1.4.36	proc_DeplGetParentWebIdFromDocUrl	77
3.1.4.36.1	Location Result Set	78
3.1.4.37	proc_DeplGetRootObjects	78
3.1.4.37.1	Root Objects Result Set	78
3.1.4.38	proc_DeplGetSystemObjects	79
3.1.4.38.1	System Objects Result Set	79
3.1.4.39	proc_DeplOrderExportObjects	79
3.1.4.40	proc_DeplReserveNextAvailableId	80
3.1.4.41	proc_DeplUpdateWebProperties	80
3.1.4.42	proc_FileExists	81
3.1.5	Timer Events	82
3.1.6	Other Local Events	82
3.2	Client Details	82
3.2.1	Abstract Data Model	83
3.2.2	Timers	83
3.2.3	Initialization	83
3.2.4	Message Processing Events and Sequencing Rules	83
3.2.4.1	Export Sequence	83
3.2.4.2	Incremental Export Sequence	85
3.2.4.3	Link Documents Export Sequence	85
3.2.4.4	Unused Web Parts Removal Sequence	86
3.2.5	Timer Events	87
3.2.6	Other Local Events	87
4	Protocol Examples	88
4.1	Export a List	88
4.2	Import a List	110
4.3	Remove Unused Web Parts After Importing a File	122
5	Security	125
5.1	Security Considerations for Implementers	125
5.2	Index of Security Parameters	125
6	Appendix A: Product Behavior	126

7 Change Tracking..... 127
8 Index 128

1 Introduction

The Windows® SharePoint® Services:Content Database Data Migration Communications protocol specifies the APIs used by a front-end Web server to retrieve objects and their meta information from the back-end database servers to do data migration. It also provides helper APIs to be used in conjunction with other protocols to import data objects into back-end database servers.

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

1.1 Glossary

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

Active Directory
Coordinated Universal Time (UTC)
GUID
principal
serialize
Unicode
XML

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

assembly
assembly name
attachment
audit flag
back-end database server
base view identifier
change log
change log identifier
checked out
content database
content type
content type identifier
current version
customizable
default form
deployment package
directory name
display name
displayed version
document
document identifier
document library
document stream
domain group
draft
event
event host
event receiver
export operation

feature
field
file
folder
folder identifier
forward link
front-end Web server
fully qualified class name
group
hyperlink
incremental export operation
incremental search scope table
leaf name
link document
list
list identifier
list item
login name
major version
member
metadict
migration system objects
minor version
mobile device
moderation status
navigation structure
page type
permission
picture library
publish
published version
publishing level
Recycle Bin
result set
return code
role
role assignment
role identifier
root folder
security group
security group identifier
security principal
security provider
security scope
server-relative URL
site
site collection
site collection identifier
site definition
site identifier
site template
stored procedure
store-relative form
subsite
SystemID

theme
Transact-Structured Query Language (T-SQL)
uncustomized
Uniform Resource Locator (URL)
user identifier
user information list
user interface (UI) version
view
Web Part
Web Part zone
Welcome page
XML document

The following terms are specific to this document:

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

1.2 References

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

1.2.1 Normative References

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

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

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

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

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

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

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

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

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

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

1.2.2 Informative References

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

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

1.3 Protocol Overview (Synopsis)

This document specifies a protocol that enables a **front-end Web server** to communicate with **back-end database servers** to perform data migration. This protocol allows a front-end Web server to retrieve data objects from the back-end database servers or to store data objects into back-end database servers. The protocol defines a number of APIs, each of which is responsible for retrieving one particular type of data object. There are certain sequence orders that the application is required to use to correctly retrieve the data objects. For example, once the data objects are retrieved from the back-end database servers, the application could **serialize** them into a file format such as the **deployment package**.

This protocol is intended to use to support the following 4 main scenarios:

1. Export all data objects from a **site**.
2. Export a list of data objects specified by the protocol client.
3. Export all data objects that have been modified since a given point in time.
4. Import data objects into back-end database servers using APIs from this protocol as well as [\[MS-WSSFO\]](#) and [\[MS-WSSDLIM\]](#).

1.4 Relationship to Other Protocols

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

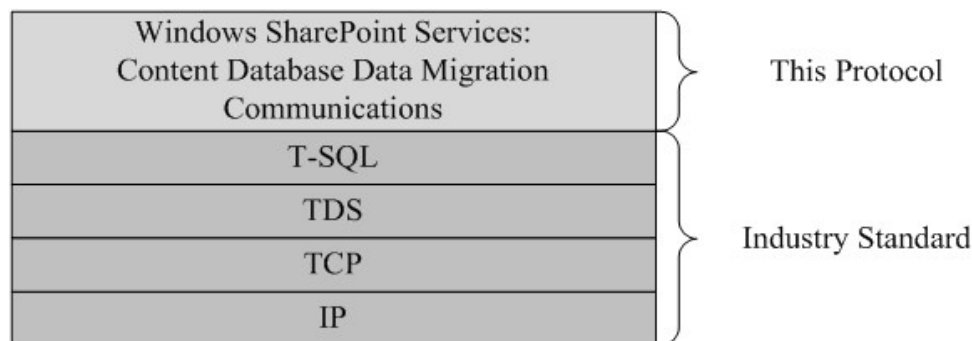


Figure 1: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

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

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

1.6 Applicability Statement

This protocol is intended to be used to import and export data objects from within the same **site collection**.

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

1.7 Versioning and Capability Negotiation

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

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

[MS-TDS] is the transport protocol used to call the stored procedures, query SQL views or SQL tables, return result codes, and return **result sets**.

2.2 Common Data Types

The following are common data types used in conjunction with this protocol.

2.2.1 Simple Data Types and Enumerations

2.2.2 Simple Data Types

2.2.2.1 Web Part Moderation

The **view** type of which user can see certain **documents** in a moderated **list**.

Value	Description
'Contributor'	System-created user's view of a moderated list, which displays a user's list items awaiting moderation in a moderated list.
'Moderator'	System-created moderator's View of a moderated list, which displays list items which are pending approval.
"	Use default behavior. If this is a moderated list, hide all unapproved list items. This value MUST be used if the list is not moderated. If this is not a moderated list, list items MUST NOT be hidden based on their moderation status .

2.2.2.2 Web Part Scope

The type of which documents are shown in the current view.

Value	Description
'Recursive'	Displays all files in the list recursively from the specified folder . Does not display folders.
'RecursiveAll'	Displays all files and folders in the list recursively from the specified folder.
'FilesOnly'	Displays all files that are immediate child files of the current folder. Does not display any list items that are Folders.
"	Displays all files and folders that are immediate child files or folders of the current folder.

2.2.3 Bit Fields and Flag Structures

2.2.3.1 Security Group Membership Flag

An integer bit mask that controls how a **security group**'s memberships are managed. The bits that are allowed are defined in the following table:

Value	Description
0x00000000	When no bits are set: Allow any user to see the security group's membership. Only the owner of this security group can change group membership. The user is not allowed to send requests to leave or join this security group. Requests to join or leave this security group are not automatically approved.
0x00000001	When bit 1 is set: Only allow members of this security group to view the group's membership information
0x00000002	When bit 2 is set: Allow members of this security group to change the group membership information
0x00000004	When bit 3 is set: Allow a user to send a request to join or leave this security group
0x00000008	When bit 4 is set: Automatically approve any user requests to join and leave this security group

2.2.4 Enumerations

2.2.4.1 Deployment Object Type

The type of data objects being exported or imported. The possible values are listed in the following table:

Value	Description
0	The Export Object is of Type Site Collection.
1	The Export Object is of Type Site.
2	The Export Object is of Type Folder.
3	The Export Object is of Type list
4	The Export Object is of Type list Item
5	The Export Object is of Type File
6	The Export Object is of Type Folder in a list.
7	The Export Object is of Type File in a list.
8	The Export Object is of Type content type
9	The Export Object is of type navigation structure
10	The Export Object is of Type field .
11	The Export Object is of type user.
12	The Export Object is of type User Changed. Used only during an incremental export operation .
13	The Export Object is of type group .
14	The Export Object is of type Group Changed. Used only during an incremental export operation.
15	The Export Object is of type role .

Value	Description
16	The Export Object is of type Role Changed. Used only during an incremental export operation.
17	The Export Object is of type role assignments .
18	The Export Object is of type Role Assignments Changed. Used only during an incremental export operation.
19	The Export Object is of type feature .
9999	The Export Object is of type Unknown.

2.2.4.2 Role Type

The type of the role. The possible values are listed in the following table:

Value	Description
0	Custom role
1	Guest role
2	Reader role
3	Contributor role
4	Web Designer role
5	Administrator role

2.2.5 Binary Structures

None.

2.2.6 Result Sets

None.

2.2.7 Tables and Views

2.2.7.1 WebParts Table

This table contains all the **Web Parts** in the back-end database servers. The WebParts Table MUST contain the following columns, in **Transact-Structured Query Language (T-SQL)**. The client uses additional columns to store other metadata about the Web parts in an implementation specific way.

```

tp_SiteId      uniqueidentifier NOT NULL,
tp_ID          uniqueidentifier NOT NULL,
tp_PageUrlID  uniqueidentifier NOT NULL,
tp_UserID     int,
tp_Level      tinyint DEFAULT 1 NOT NULL,

```

tp_SiteId: the **site collection identifier** of the site collection that contains this Web part.

tp_ID: the identifier of the Web part.

tp_PageUrlId: The **document identifier** of the document that contains this Web part

tp_UserID: When this value is set to NULL, it indicates that this Web part is visible to everyone. If the value is not NULL, it indicates that this Web part is a personal Web part, and is visible only to the user whose identifier equals this value.

tp_Level: **publishing level** value of the Web part

2.2.7.2 WebPartLists Table

Each role in this table represents a Web part in a list that is stored in the back-end database servers. This table MUST contain the following columns, in T-SQL. The client uses additional columns to store other metadata about the Web parts in an implementation specific way.

```
tp_SiteId    uniqueidentifier NOT NULL,  
tp_Level    tinyint DEFAULT 1 NOT NULL
```

tp_SiteId: the site collection identifier of the site collection that contains this Web part.

tp_Level: Publishing level value of the Web part.

2.2.7.3 #WebParts Table

This is a temporary table created on the Back-end database server to store the identifiers of the Web parts that are referenced by a File that is being imported. The #WebParts table is defined using T-SQL syntax as follows:

```
TABLE #WebParts(  
    WebPartId    uniqueidentifier NOT NULL PRIMARY KEY  
);
```

WebPartId: Identifier of a Web part that is referenced by a file being imported.

2.2.8 XML Structures

No common XML structures are defined in this protocol.

2.2.8.1 Namespaces

None.

2.2.8.2 Simple Types

None.

2.2.8.3 Complex Types

None.

2.2.8.4 Elements

None.

2.2.8.5 Attributes

None.

2.2.8.6 Groups

None.

2.2.8.7 Attribute Groups

None.

3 Protocol Details

3.1 Server Details

3.1.1 Abstract Data Model

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

The server stores a hierarchy of objects. Operations that can be performed against the server are divided into 2 main categories: export and import. During an **export operation**, the server maintains a table of export objects and in some cases an **incremental search scope table**. These tables are used to keep track of what objects MUST be included in the export operation. During an import operation, the server APIs in this protocol are stateless.

3.1.2 Timers

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

3.1.3 Initialization

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

3.1.4 Message Processing Events and Sequencing Rules

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

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

3.1.4.1 proc_DeplAddExportObject

The `proc_DeplAddExportObject` Stored Procedure is called during an Export Operation to add a data object to the table of export objects. The `proc_DeplAddExportObject` is defined using T-SQL syntax, as follows:

```
PROCEDURE proc_DeplAddExportObject (
    @SiteId          uniqueidentifier,
    @Id              uniqueidentifier,
    @Url             nvarchar(260),
    @Type            tinyint,
    @IncludeDescendants tinyint,
```

```

@IsMajorFileOnly      int,
@IsLink                tinyint,
@IsSelected            tinyint
);

```

@SiteId: The site collection identifier for the **Site Collection** that contains the object.

@Id: The data object to be exported. If this parameter is NULL, the @Url parameter will be used to determine the data object. The exact meaning of this parameter is determined by the @Type parameter. For example, if the @Type is 1, then @Id is the **site identifier** of a site, if the @Type is 3, then @Id is the **list identifier** of a list, and so on.

@Url: The **store-relative form** of the object to be exported. If this parameter is NULL, the @Id parameter will be used to determine the data object. If both @Id and @Url are not NULL, @Url will be used.

@Type: An 8 bit integer value that specifies the type of the data object to be exported. The parameter is only meaningful when either @Id or @Url is not NULL. It MUST be one of the following values:

Value	Description
1	The data object is a site.
2	The data object is a folder .
3	The data object is a list.
4	The data object is a list item .
5	The data object is a document .

@IncludeDescendants: This parameter is an 8-bit integer value that specifies whether the child objects of this object MUST be included in the Export Operation. This Stored Procedure will save this value in the table of export objects. When [proc_DeplCalculateChildrenToExport](#), specified in section 3.1.4.6, is called, it reads this value and adds the child objects of this object to the table of export objects if necessary. The settings only affect objects with @Type of 1, 2, or 3. This parameter MUST be one of the following values:

Value	Description
0	When @Type is 2, subfolders and their descendants MUST NOT be exported. When @Type is 1, subsites MUST NOT be exported. When @Type is 3, list items from the list MUST NOT be exported.
1	When @Type is 2, subfolders and their descendants MUST NOT be exported. When @Type is 1, Subsites MUST NOT be exported. When @Type is 3, list items from the list MUST be exported.
2	When @Type is 2, subfolders and their descendants MUST be exported. When @Type is 1, Subsites MUST be exported. When @Type is 3, list items from the list MUST be exported.

@IsMajorFileOnly: This parameter MUST be either 0 or 1. When set to 1, this stored procedure will only add the data object to the table of export objects if it is a **major version**. Otherwise, this stored procedure will add the data object if it is either a major version or **minor version**. It only takes effect when the @Type parameter is of value 4 or 5.

@IsLink: This parameter MUST be either 0 or 1. When set to 1, this parameter indicates that the object being exported was a **link document**. Otherwise, the object is not a link document. It only takes effect when the @Type parameter is of value 2, 4, or 5.

@IsSelected: This parameter MUST be either 0 or 1. When set to 0, it indicates that the object is being exported as part of Export Operation of a full **Site**. Otherwise, it indicates that the client has chosen to export a list of objects smaller than a site, including this one.

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

Value	Description
0	Execution finished. The objects have been successfully appended to the table of export objects.
2	This value MUST be returned if the object specified by the @Url parameter could not be found, or if @SiteId is NULL. This value SHOULD be returned if @id specifies an empty GUID .<1>
87	Invalid parameter. This value MUST be returned if the @Id is NULL, or @Type does not have a value specified in its description

Result Sets: MUST NOT return any result sets.

3.1.4.2 proc_DeplAddExportObjectDependencies

The **proc_DeplAddExportObjectDependencies** Stored Procedure is called during an export operation to add additional, dependent objects that are necessary to correctly persist the state of the objects. The following objects are considered dependency objects:

1. If the object is a site: the **root folder** of the site, the **migration system objects** (which includes "Lists", "_cts", "_private" and "_catalogs"), the catalog lists (which are lists created by the system), Fields and content types, **Welcome pages**, navigation structures, roles and role assignments are dependent objects.
2. If the object is a list: the Root Folder and Documents in the list (including the template Documents) are dependent objects.
3. If the object is a folder: the list item associated with this folder is a dependent object
4. If the object is a document: the list item associated with this document is a dependent object
5. If the object is a list Item: the Document and Content Types associated with this list item are dependent objects.

proc_DeplAddExportObjectDependencies is defined using T-SQL syntax, as follows:

```
PROCEDURE proc_DeplAddExportObjectDependencies (  
    @SiteId                uniqueidentifier,  
    @IsMajorFileOnly      int,  
    @LinksOnly            tinyint  
);
```

@SiteId: The site collection identifier for the **Site Collection** whose content will be exported.

@IsMajorFileOnly: This parameter MUST be either 0 or 1. When set to 1, this Stored Procedure will only add Documents whose latest versions are major versions. When set to 0, this Stored Procedure will add Documents whose latest versions are either major versions or minor versions.

@LinksOnly: This parameter MUST be either 0 or 1. When set to 1, this Stored Procedure will only add dependencies for link documents in the table of export objects. When set to 0, this Stored Procedure will add dependencies for non link documents in the table of export objects.

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

Value	Description
0	Execution finished.
87	Invalid Parameter. This is returned when @SiteId IS NULL

Result Sets: MUST NOT return any result sets.

3.1.4.3 proc_DeplAddExportObjectLinks

The proc_DeplAddExportObjectLinks Stored Procedure is called during an Export Operation to add additional Documents to the table of export objects. The additional Documents are those which are referenced (in their content) by items already in the table of export objects. This Stored Procedure will only add Documents that belong to the same Site Collection as the Documents making the reference. proc_DeplAddExportObjectLinks is defined using T-SQL syntax, as follows:

```
PROCEDURE proc_DeplAddExportObjectLinks (
    @SiteId          uniqueidentifier,
    @IsMajorFileOnly tinyint
);
```

@SiteId: The site collection identifier for the **Site Collection** whose content will be exported.

@IsMajorFileOnly: This parameter MUST be either 0 or 1. When set to 1, this Stored Procedure will only add Documents whose latest versions are major versions. When set to 0, this Stored Procedure will add Documents whose latest versions are either major versions or minor versions.

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

Value	Description
0	Execution finished.
87	Invalid parameter value. @SiteId is NULL

Result Sets: MUST return the following Result Set:

3.1.4.3.1 Link Count Result Set

The Link Count Result Set contains at most one row. The Result Set returns the number of Documents it finds and adds them to the list of objects that will be exported. If @SiteId cannot be found or does not exist, the Result Set will still return one row with {LinkCount} =0.

The Link Count Result Set is defined using T-SQL syntax, as follows:

```
{LinkCount}          int;
```

{LinkCount}: Specifies the number of Documents found and added to the table of export objects.

3.1.4.4 proc_DeplAddIncrementalSearchScope

The proc_DeplAddIncrementalSearchScope Stored Procedure is called during an Export Operation to add a Site to the Incremental Search Scope Table.

proc_DeplAddIncrementalSearchScope is defined using T-SQL syntax, as follows:

```
PROCEDURE proc_DeplAddIncrementalSearchScope (  
    @SiteId                uniqueidentifier,  
    @Id                    uniqueidentifier,  
    @Url                   nvarchar(260),  
    @Type                  tinyint,  
    @IncludeDescendants     tinyint,  
    @IsMajorFileOnly      int,  
    @ChangeTimeStart      datetime,  
    @ChangeNumberStart    int;  
);
```

@SiteId: The site collection identifier for the **Site Collection** that contains the data objects to be exported.

@Id: The **Site** that is being added to the Incremental Search Scope Table.

@Url: The **Store-Relative Form** of the Site. If this parameter is NULL, the Stored Procedure MUST look up the **URL** of the Site using the @Id parameter.

@Type: This MUST be 1.

@IncludeDescendants: Refer to the @IncludeDescendants parameter of [proc_DeplAddExportObject](#) for a complete list of acceptable values of this parameter.

@IsMajorFileOnly: Reserved. This parameter's value MUST be ignored.

@ChangeTimeStart: This parameter specifies a time stamp in **Coordinated Universal Time (UTC)**. Only changes made after this time will be included in the Export Operation.

@ChangeNumberStart: This parameter specifies a **change log** identification number. Only changes with a larger identification number than this value will be included.

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

Value	Description
0	Execution finished.
2	Cannot find a Site with site identifier equal to the specified @Id, or a Store-Relative Form equal to the specified @Url
87	Invalid parameter. This is returned when the @Type is not 1. Or both @Id and @Url are NULL

Result Sets: MUST NOT return any result sets.

3.1.4.5 proc_DeplAddSite

The proc_DeplAddSite Stored Procedure is called during an **Export Operation** to add a **Site Collection** whose data objects will be exported to the table of export objects. This Stored Procedure will not add any actual data objects. The data objects are added by calling

[proc_DeplAddExportObject](#), specified in section 3.1.4.1. `proc_DeplAddSite` is defined using T-SQL syntax, as follows:

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

@SiteId: The site collection identifier for the **Site Collection** whose data objects will be exported.

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

Value	Description
0	Execution finished.
87	Invalid parameter value. @SiteId is NULL

Result Sets: MUST NOT return any result sets.

3.1.4.6 `proc_DeplCalculateChildrenToExport`

The `proc_DeplCalculateChildrenToExport` Stored Procedure is called to recursively expand the table of export objects to include the child objects of the objects that are currently stored in the table of export objects. The following objects are considered child objects:

- If the object is a Folder, all Documents and Subfolders are considered child objects.
- If the object is a Site, all the Subsites are child objects of this Site. Additionally, the Navigation Structure of this Site, the roles and role assignments of this Site, and all the lists in this Site are considered child objects.
- If the object is a Document that belongs to a **document library**, the list items associated with this Document is considered child object.

`proc_DeplCalculateChildrenToExport` is defined using T-SQL syntax, as follows:

```
PROCEDURE proc_DeplCalculateChildrenToExport (  
    @SiteId                uniqueidentifier,  
    @IsMajorFileOnly      int,  
    @LinksOnly             tinyint,  
    @IncludeUserInfo      bit  
)  
;
```

@SiteId: The site collection identifier for the **Site Collection** whose content is being exported.

@IsMajorFileOnly: This parameter MUST be either 0 or 1. When set to 1, this Stored Procedure will only add Documents whose last version is major versions to the table of export objects. Otherwise, this Stored Procedure will add Documents whose last version is either major or minor.

@LinksOnly: This parameter MUST be either 0 or 1. When set to 1, this Stored Procedure will only add child objects for link documents in the table of export objects. When set to 0, this Stored Procedure will add child objects for non link documents in the table of export objects.

@IncludeUserInfo: This parameter MUST be either 0 or 1. When set to 1, this Stored Procedure will allow lists, Folders and list items from the **user information list** to be included as child objects. When set to 0, those objects will not be included as child objects.

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

Value	Description
0	Execution finished.
87	Invalid Parameter. This code is returned when @SiteId is NULL

Result Sets: MUST NOT return any result sets.

3.1.4.7 proc_DeplCalculateIncrementalSearchScope

The proc_DeplCalculateIncrementalSearchScope Stored Procedure is called to add Subsites into an Incremental Search Scope Table, if necessary. The Incremental Search Scope Table is created by calling [proc_DeplAddIncrementalSearchScope](#), specified in section 3.1.4.4. The @IncludeDescendants parameter passed to proc_DeplAddIncrementalSearchScope decides whether Subsites MUST be included for a Site in the Incremental Search Scope Table when this Stored Procedure is called.

proc_DeplCalculateIncrementalSearchScope is defined using T-SQL syntax, as follows:

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

@SiteId: The site collection identifier for the **Site Collection** that contains the Sites in the Incremental Search Scope Table.

Return Code Values: proc_DeplCalculateIncrementalSearchScope returns an integer which MUST be listed in the following table:

Value	Description
0	Execution finished.
87	Invalid Parameter. This is returned when the @SiteId is NULL.

Result Sets: proc_DeplCalculateIncrementalSearchScope MUST NOT return any result sets.

3.1.4.8 proc_DeplExportGroups

The proc_DeplExportGroups Stored Procedure is called to retrieve information about one Security Group from the back-end database servers. To retrieve all security groups from a Site Collection, a client application calls this Stored Procedure with @GroupIdCur = -1, and then calls this Stored Procedure iteratively, each time with a bigger @GroupIdCur parameter, by using the value in the ID column of the [Security Group Information result set](#), specified in section 3.1.4.8.1, from the previous call.

proc_DeplExportGroups is defined using T-SQL syntax, as follows:


```

PROCEDURE proc_DeplExportGroups (
    @SiteId          uniqueidentifier,
    @GroupIdCur     int
);

```

@SiteId: The site collection identifier for the **Site Collection** which contains the Security Group.

@GroupIdCur: A Security Group identifier. This Stored Procedure will only return the next Security Group whose identifier is larger than this value, in sorted order.

Return Code Values: An integer which the protocol client MUST ignore.

Result Sets: MUST return 2 result sets in the following order:

3.1.4.8.1 Security Group Information Result Set

The Security Group Information Result Set returns information about a Security Group and MUST contain at most one row.

The Security Group Information Result Set is defined using T-SQL syntax, as follows:

```

ID                int,
Title             nvarchar(255),
Description       nvarchar(512),
Owner            int,
OwnerIsUser      bit,
RequestEmail     nvarchar(255),
Flags            int;

```

ID: The identifier of a Security Group

Title: The name of this Security Group

Description: The description of this Security Group

Owner: The **security group identifier** of the owner of this Security Group. This could be either a security principal identifier, or a Security Group identifier. The value of OwnerIsUser MUST be used to determine which one is the case.

OwnerIsUser: This MUST be either 0 or 1. When set to 1, the owner of this Security Group is a **security principal**. When set to 0, the owner is any member from this Security Group.

RequestEmail: The e-mail address where a user could send a request to join or leave the Security Group.

Flags: A value specified in section 2.2.2.1, [Security Group Membership Flag](#).

3.1.4.8.2 Security Group Memberships Result Set

The Security Group Memberships Result Set returns the list of security principals that are members of this Security Group. This Result Set MUST contain the number of rows equals to the number of the members in this Security Group. If the Security Group contains no members, this Result Set MUST return 0 rows.

The Security Group Memberships Result Set is defined using T-SQL syntax, as follows:

```
MemberId int;
```

MemberId: The security principal identifier of this member.

3.1.4.9 proc_DeplExportGroupX

The proc_DeplExportGroupX Stored Procedure is called during an incremental Export Operation to retrieve information about changes made to a security Group.

proc_DeplExportGroupX is defined using T-SQL syntax, as follows:

```
PROCEDURE proc_DeplExportGroupX(  
    @SiteId          uniqueidentifier,  
    @EventTime       datetime,  
    @ChangeLogId     bigint  
);
```

@SiteId: The site collection identifier for the **Site Collection** containing security groups.

@EventTime: The time stamp, in UTC, of an **Event** from the Change Log which records modification made to this security Group.

@ChangeLogId: The **change log identifier** of the same Event as described in @EventTime.

Return Code Values: An integer which the protocol client MUST ignore.

Result Sets: MUST return the following Result Set:

3.1.4.9.1 Security Group Change Result Set

The Security Group Change Result Set returns information about changes made to a security Group. The Security Group Change Result Set will return one row if the Event specified by @ChangeLogId is found. Otherwise, the Result Set will contain 0 rows.

The Security Group Change Result Set is defined using T-SQL syntax, as follows:

```
EventType      int,  
ID             int,  
Title          nvarchar(255),  
Description    nvarchar(512),  
tp_Login       nvarchar(255),  
OwnerIsUser   bit,  
tp_Id         int,  
tp_Login       nvarchar(255),  
RequestEmail  nvarchar(255),  
Flags         int;
```

EventType: An Event Type Flag.

ID: The security Group identifier.

Title: The name of this security Group.

Description: The description of this security Group.

tp_Login: The **login name** of the security principal which is the owner of this security Group. If the owner of this security Group is not a security principal, this value MUST be NULL.

OwnerIsUser: This MUST be either 0 or 1. When set to 1, the owner of this security Group is a security principal. When set to 0, any Member from this security Group is an owner.

tp_Id: An integer identifier of the security principal that is added to or removed from the security Group. When **{EventType}** = 2097152, this value represents the security principal to add to the security Group. When **{EventType}** = 4194304, this value represents the security principal to remove from the security Group. For any other value of **{EventType}**, this value MUST be NULL.

tp_Login: The login name of the security principal whose identifier is **{tp_Id}**. This value MUST be NULL if **{tp_Id}** is NULL.

RequestEmail: The e-mail address where a user could send a request to join or leave the security Group.

Flags: A value specified in section 2.2.2.1, [Security Group Membership Flag](#).

3.1.4.10 proc_DeplExportRoleAssignments

The proc_DeplExportRoleAssignments Stored Procedure is called during an Export Operation to retrieve a list of role assignments from a Site. proc_DeplExportRoleAssignments is defined using T-SQL syntax, as follows:

```
PROCEDURE proc_DeplExportRoleAssignments(  
    @SiteId                uniqueidentifier,  
    @WebId                 uniqueidentifier,  
    @WssGroupOnly         bit,  
    @ScopeIdCur          uniqueidentifier  
);
```

@SiteId: The site collection identifier for the **Site Collection** which contains the role assignments to be retrieved.

@WebId: The site identifier for the Site from which the role assignments will be retrieved.

@WssGroupOnly: When set to 1, this Stored Procedure will only retrieve the role assignments in which a role is assigned to a security Group. Otherwise, it will retrieve all role assignments.

@ScopeIdCur: This parameter represents a **security scope**. This Stored Procedure MUST only return a role assignment for a security Scope whose value is larger than this value, in sorted order.

Return Code Values: An integer which the protocol client MUST ignore.

Result Sets: MUST return two result sets in the following order:

3.1.4.10.1 Scope Information Result Set

The Scope Information Result Set returns information about a security Scope where role assignments are defined. This Result Set MUST return at most one row. If any of the parameters are incorrectly specified, this Result Set MUST return 0 rows.

The Scope Information Result Set is defined using T-SQL syntax, as follows:

```
SiteId                uniqueidentifier,
```

```

WebId                uniqueidentifier,
ScopeId              uniqueidentifier,
RoleDefWebId         uniqueidentifier,
FullUrl              nvarchar(260),
{ObjectId}           uniqueidentifier,
{ObjectType}         int,
{ScopeUrl}           nvarchar(256),
AnonymousPermMask    bigint;

```

SiteId: The site collection identifier for the **Site Collection** that contains the **security Scope**.

WebId: The site identifier for the **Site** that contains the security scope.

ScopeId: The identifier of the security scope.

RoleDefWebId: The site identifier for the Site that contains the definitions of the roles that are used in this security scope.

FullUrl: The **Store-Relative Form** of the Site that contains the definitions of the roles that are used in this **security scope**.

{ObjectId}: The root object of the security scope. The value of @ObjectType determines the type of this value.

{ObjectType}: The type of the root object of the security scope as defined in the following table:

Value	Description
0	The object is a Site.
1	The object is a list.
2	The object is a Document or list item

{ScopeUrl}: The Store-Relative Form of the root object of the security scope.

AnonymousPermMask: A 64-bit mask that specifies the **Permissions** this role allows for an anonymous user. The values of this field are specified in [\[MS-WSSF0\]](#) section 2.2.2.13 WSS Rights Mask.

3.1.4.10.2 Role Assignment Information Result Set

The Role Assignment Information Result Set returns a list of role assignments. The number of rows in the Role Assignment Information Result Set is equal to the number of role assignments defined on the security scope returned by the Role Assignment Information Result Set. When @WssGroupOnly is 1, the number of rows returned will be reduced to only the role assignments in which a role is assigned to a Security Group.

The Role Assignment Information Result Set is defined using T-SQL syntax, as follows:

```

RoleId              int,
PrincipalId         int,
tp_Id              int;

```

RoleId: The **role identifier**.

PrincipalId: The security principal identifier.

tp_Id: The same Security principal identifier as **{PrincipalId}** column, when **@WssGroupOnly = 0**. When **@WssGroupOnly = 1**, this value SHOULD [<2>](#) be NULL.

3.1.4.11 proc_DeplExportRoleAssignmentX

The proc_DeplExportRoleAssignmentX Stored Procedure is called during an Incremental **Export Operation** to retrieve information about an Event in which one or more role assignments are modified. proc_DeplExportRoleAssignmentX is defined using T-SQL syntax, as follows:

```
PROCEDURE proc_DeplExportRoleAssignmentX(  
    @SiteId            uniqueidentifier,  
    @EventTime         datetime,  
    @ChangeLogId       bigint  
);
```

@SiteId: The site collection identifier for the **Site Collection** in which the Event has occurred.

@EventTime: The time, in UTC, when this Event occurred.

@ChangeLogId: The Change Log identifier of this Event.

Return Code Values: An integer which the protocol client MUST ignore.

Result Sets: If there is an Event specified by @EventTime and @ChangeLogId, then the following Result Set MUST be returned.

Otherwise, Result Set MUST NOT be returned.

3.1.4.11.1 Role Assignment Change Result Set

The Role Assignment Change Result Set returns information about the changes that have occurred to one or more role assignment. This Result Set MUST return at most one row. The Role Assignment Change Result Set is defined using T-SQL syntax, as follows:

```
{EventType}            int,  
{OpCode}               int  
{WebId}                uniqueidentifier,  
{ScopeId}              uniqueidentifier,  
{RoleDefWebId}         uniqueidentifier,  
{RoleDefWebFullUrl}   nvarchar(256),  
{ScopeObjectId}       uniqueidentifier,  
{ScopeObjectType}     int,  
{ScopeUrl}             nvarchar(256),  
{AnonymousPermMask}   bigint,  
{RoleTitle}            nvarchar(255),  
{GroupId}              int,  
{GroupTitle}           nvarchar(255),  
{UserId}               int,  
{UserLogin}            nvarchar(255);
```

{EventType}: An Event Type Flag.

{OpCode}: A Security Change Type Flag, specified in [\[MS-WSSDLIM\]](#) section 2.2.2.3 Security Change Type Flags.

{WebId}: The Site in which this change has taken place.

{ScopeId}: The identifier of the security scope in which this Event has occurred.

{RoleDefWebId}: The Site that contains the definitions of the roles that are used in the security scope defined by {ScopeId}.

{RoleDefWebFullUrl}: The Store-Relative Form of the Site that contains the definitions of the roles that are used in the security scope defined by {ScopeId}.

{ScopeObjectId}: The root object of the security scope defined by {ScopeId}.

{ScopeObjectType}: The type of the object specified by {ScopeObjectId}. The value of this parameter MUST be one of the following:

Value	Description
0	The object is expected to be a Site.
1	The object is expected to be a list.
2	The object is expected to be a Document or list item.

{ScopeUrl}: The Store-Relative Form of the root object of the security scope defined by {ScopeId}.

{AnonymousPermMask}: A 64-bit mask that specifies the permissions granted to anonymous Users in the security scope defined by {ScopeId}. The bit mask values are specified in [\[MS-WSSFO\]](#) section 2.2.2.13 WSS Rights Mask.

{RoleTitle}: The name of the role whose assignment has changed.

{GroupId}: The Security Group identifier of a security group if a Security Group is assigned to the role specified by {RoleTitle} during this change. Otherwise, this value MUST be NULL.

{GroupTitle}: The name of the Security Group if it is assigned to the role (as specified by {RoleTitle}) during this change. Otherwise, this value MUST be NULL.

{UserId}: The Security principal identifier if a security principal is assigned to the role (as specified by {RoleTitle}) during this change. Otherwise, the value MUST be NULL.

{UserLogin}: The Login Name of the security principal if it is assigned to the role specified by {RoleTitle} during this change. Otherwise, this value MUST be NULL.

3.1.4.12 proc_DeplExportRoles

The proc_DeplExportRoles Stored Procedure is called to retrieve a list of roles from a Site.

proc_DeplExportRoles is defined using T-SQL syntax, as follows:

```
PROCEDURE proc_DeplExportRoles(  
    @SiteId          uniqueidentifier,  
    @WebId           uniqueidentifier,  
    @RoleIdCur     int  
)  
;
```

@SiteId: The site collection identifier for the **Site Collection** from which the roles are being retrieved.

@WebId: The site identifier for the Site where the roles are defined.

@RoleIdCur: The role identifier. This Stored Procedure will only return roles whose IDs are larger than this value, in sorted order.

Return Code Values: An integer which the protocol client MUST ignore.

Result Sets: MUST return the following Result Set:

3.1.4.12.1 Role Information Result Set

The Role Information Result Set returns information of a list of roles. This Result Set will return 100 rows, or one row for each role, whichever is smaller. Each row contains information about a role. The Role Information Result Set is defined using T-SQL syntax, as follows:

```
SiteId          uniqueidentifier,  
Id              uniqueidentifier,  
RoleId          int,  
Title           nvarchar(255),  
Description     nvarchar(512),  
PermMask       bigint,  
Hidden         bit,  
RoleOrder      int,  
Type           tinyint;
```

SiteId: The site collection identifier of the Site Collection where this role is retrieved.

Id: The site identifier of the Site where this role is defined.

RoleId: The role identifier.

Title: The name of the role.

Description: The description of the role.

PermMask: A 64-bit mask that specifies the permissions that this role allows. The bit mask values are specified in [\[MS-WSSFO\]](#) section 2.2.2.13 WSS Rights Mask.

Hidden: MUST be either 0 or 1. When set to 1, this role MUST be hidden in the UI.

RoleOrder: This value MUST be used to determine the order of which roles are represented in the UI.

Type: The [Role Type](#), specified in section 2.2.3.2.

3.1.4.13 proc_DeplExportRoleX

The proc_DeplExportRoleX Stored Procedure is called during an Incremental Export Operation to retrieve information about an Event in which a role is modified. proc_DeplExportRoleX is defined using T-SQL syntax, as follows:

```
PROCEDURE proc_DeplExportRoleX(  
    @SiteId          uniqueidentifier,
```

```

    @EventTime          datetime,
    @ChangeLogId       bigint
);

```

@SiteId: The site collection identifier for the **Site Collection** in which this Event is being retrieved.

@EventTime: This value, in UTC, specifies when this Event has occurred.

@ChangeLogId: The Change Log identifier of this Event in the Change Log.

Return Code Values: An integer which the protocol client MUST ignore.

Result Sets: MUST return the following Result Set:

3.1.4.13.1 Role Change Result Set

The Role Change Result Set returns information about changes that have occurred to a role. This Result Set MUST return one row in the Result Set, if the Event specified by the @EventTime and @ChangeLogId can be found, or zero rows if the Event could not be found. The Role Change Result Set is defined using T-SQL syntax, as follows:

```

EventType          int,
{Operation Code}   int,
Guid0              uniqueidentifier,
Int0               int,
WebId              uniqueidentifier,
FullUrl            nvarchar(256),
RoleId             int,
Title              nvarchar(255),
ItemName           nvarchar(255),
Description         nvarchar(512),
PermMask           bigint,
Hidden             bit,
RoleOrder          int,
Type               tinyint,
tp_Id              int,
tp_Login           nvarchar(255);

```

EventType: An Event Type Flag, specified in [\[MS-WSSDLIM\]](#) section 2.2.2.2 Event Type Flags.

{Operation Code}: A Security Change Type Flag, specified in [\[MS-WSSDLIM\]](#) section 2.2.2.3 Security Change Type Flags.

Guid0: The identifier of the security scope where this change occurred.

Int0: If this value is NOT NULL, it MUST contain the identifier of the security principal that created the role and the Event specified by @EventTime and @ChangeLogId MUST be an Event in which a role is created, and the EventType column MUST have bit 19 (value = 262144) set. If this value is NULL, it MUST be ignored.

WebId: The Site identifier for the Site where this role is defined.

FullUrl: The Store-Relative Form URL of the Site where this role is defined.

RoleId: A role identifier, if EventType does not have bit 24 (value = 8388608) set. Otherwise, this value MUST be NULL

Title: The name of the role if RoleId is NOT NULL. Otherwise this value MUST be NULL.

ItemName: The name of the role if RoleId is NULL. This happens when a role is deleted. Otherwise, this value MUST be NULL.

Description: The description of the role if RoleId is NOT NULL. Otherwise this value MUST be NULL.

PermMask: A 64-bit mask that specifies the permissions this role allows. If RoleId is NOT NULL, the bit mask's values are specified in [\[MS-WSSFO\]](#) section 2.2.2.13 WSS Rights Mask. Otherwise this value MUST be NULL

Hidden: This value MUST be either 0 or 1. When RoleId is NOT NULL, this value specifies whether this role MUST be hidden in the UI. Otherwise this value MUST be NULL.

RoleOrder: This value MUST be used to determine the order of which roles are represented in the UI if RoleId is NOT NULL. Otherwise this value MUST be NULL.

Type: The [Role Type](#), specified in section 2.2.3.2. Otherwise this value MUST be NULL.

tp_Id: The same value as Int0, if Int0 is NOT NULL. Otherwise, this value MUST be NULL.

tp_login: The Login Name of the security principal represented by Int0, if Int0 is NOT NULL. Otherwise, this value MUST be NULL.

3.1.4.14 proc_DeplExportUsers

The proc_DeplExportUsers Stored Procedure is called to retrieve a list of security principals from the back-end database servers. Each time this Stored Procedure is called, it returns at most 100 security principals. This Stored Procedure MUST be called additional times to retrieve more security principals.

proc_DeplExportUsers is defined using T-SQL syntax, as follows:

```
PROCEDURE proc_DeplExportUsers(  
    @SiteId          uniqueidentifier,  
    @UserIdCur      int  
)  
;
```

@SiteId: The site collection identifier for the **Site Collection** that contains the security principals to be retrieved.

@UserIdCur: A security principal identifier. This Stored Procedure MUST only return security principals whose IDs are larger than this value, in sorted order. The application MUST use -1 the first time it calls this Stored Procedure.

Return Code Values: An integer which the protocol client MUST ignore.

Result Sets: MUST return the following Result Set:

3.1.4.14.1 Security Principal Information Result Set

The Security Principal Information Result Set returns information about a list of security principals. This Result Set returns 100 rows, or one row for each security principal, whichever is smaller. Each row contains information about one security principal. The Security Principal Information Result Set is defined using T-SQL syntax, as follows:

```

tp_SiteId          uniqueidentifier,
tp_Id              int,
tp_Title           nvarchar(255),
tp_Login           nvarchar(255),
tp_Email           nvarchar(255),
tp_Notes           nvarchar(1023),
tp_DomainGroup     bit,
tp_SiteAdmin       bit,
tp_SystemId        varbinary(512),
tp_Deleted         int;

```

tp_SiteId: The site collection identifier for the Site Collection that contains this security principal.

tp_Id: The security principal identifier.

tp_Title: The **display name** of the security principal. This value MUST NOT be NULL

tp_Login: The Login Name of the security principal. This value MUST NOT be NULL

tp_Email: The e-mail address of the security principal. This value MUST NOT be NULL

tp_Notes: A general description of the security principal. This value MUST NOT be NULL

tp_DomainGroup: This MUST be either 0 or 1. When set to 1, it indicates that the current security principal is a **domain group** from the **Active Directory**.

tp_SiteAdmin: This MUST be either 0 or 1. When set to 1, it indicates that the current security principal is an administrator of the containing Site Collection.

tp_SystemId: The **SystemID** of the security principal.

tp_Deleted: A value specifying whether the security principal is marked as "deleted" within this Site Collection. If tp_Deleted is 0, the security principal is not deleted. If tp_Deleted is not 0, the Principal is marked as deleted and the value is the same as the security principal identifier. The Deleted state MUST be kept in the back-end database servers, rather than deleting the security principal completely, 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.

3.1.4.15 proc_DeplExportUserX

The proc_DeplExportUserX Stored Procedure is called during an Incremental Export Operation to retrieve information about an Event in which a security principal is modified. proc_DeplExportUserX is defined using T-SQL syntax, as follows:

```

PROCEDURE proc_DeplExportUserX(
    @SiteId          uniqueidentifier,
    @EventTime       datetime,
    @ChangeLogId     bigint
);

```

@SiteId: The site collection identifier for the **Site Collection** in which the Event has occurred.

@EventTime: The time, in UTC, when the Event has occurred.

@ChangeLogId: The Change log identifier of this Event.

Return Code Values: An integer which the protocol client MUST ignore.

Result Sets: MUST return the following Result Set:

3.1.4.15.1 Security Principal Change Result Set

The Security Principal Change Result Set returns information about changes that have occurred to a security principal. This Result Set will be returned if there is a Change Log Event that matches the @EventTime and @ChangeLogId parameter and the object modified in that event was a security principal. This Result Set contains at most one row.

The Security Principal Change Result Set is defined using T-SQL syntax, as follows:

```
EventType          int,  
tp_Id              int,  
tp_Title           nvarchar(255),  
tp_Login           nvarchar(255),  
tp_Email           nvarchar(255),  
tp_Notes           nvarchar(1023),  
tp_DomainGroup    bit,  
tp_SiteAdmin       bit,  
tp_SystemId       varbinary(512);
```

EventType: An Event Type Flag, specified in [\[MS-WSSDLIM\]](#) section 2.2.2.2 Event Type Flags.

tp_Id: The security principal identifier.

tp_Title: The display name of the security principal. This value MUST NOT be NULL.

tp_Login: The Login Name of the security principal. This value MUST NOT be NULL.

tp_Email: The e-mail address of the security principal. This value MUST NOT be NULL.

tp_Notes: A general description of the security principal. This value MUST NOT be NULL.

tp_DomainGroup: This MUST be either 0 or 1. When set to 1, it indicates that the current security principal is a **Domain Group** from the **Active Directory**.

tp_SiteAdmin: This MUST be either 0 or 1. When set to 1, it indicates that the current security principal is an administrator of the containing Site Collection.

tp_SystemId: The SystemID of the security principal.

3.1.4.16 proc_DeplFindScopeObject

The proc_DeplFindScopeObject Stored Procedure is called to find an object given its **Store-Relative Form** and Type. The object returned by the output parameter @ObjectId can be used to import Role Assignments during an Import Operation. proc_DeplFindScopeObject is defined using T-SQL syntax, as follows:

```
PROCEDURE proc_DeplFindScopeObject (  
    @SiteId          uniqueidentifier,  
    @ObjectUrl       nvarchar(260),  
    @ObjectType      int,  
    @ObjectId        uniqueidentifier OUTPUT
```

);

@SiteId: The site collection identifier for the **Site Collection** in which the object is being looked up.

@ObjectUrl: The Store-Relative Form of the object that is being looked up.

@ObjectType: The expected type of the object. The value of this parameter MUST be one of the following:

Value	Description
0	The object is expected to be a Site.
1	The object is expected to be a list.
2	The object is expected to be a Document or list item.

@ObjectId: This is an OUTPUT parameter. If the return value is 0, this parameter will hold the identifier of the object, as specified in @ObjectType.

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

Value	Description
0	Execution finished.
2	File not found. This is returned when the @SiteId and @ObjectUrl does not specify a valid object, or the object is not the same type as @ObjectType.

Result Sets: MUST NOT return any result sets.

3.1.4.17 **proc_DeplFindScopeObjectById**

3.1.4.18 **proc_DeplFixWebPartsOnListCreation**

The `proc_DeplFixWebPartsOnListCreation` Stored Procedure is called during an Import Operation to associate a **Web Part** with a list that has just been created. This Stored Procedure updates the Web Part information with a new list identifier.

`proc_DeplFixWebPartsOnListCreation` is defined using T-SQL syntax, as follows:

```
PROCEDURE proc_DeplFixWebPartsOnListCreation (  
    @SiteId                uniqueidentifier,  
    @WebPartId             uniqueidentifier,  
    @OldListId             uniqueidentifier,  
    @NewListId             uniqueidentifier  
);
```

@SiteId: The site collection identifier for the **Site Collection** that contains the Web Part and the list.

@WebPartId: The Identifier of the Web Part being updated.

@OldListId: The list identifier of the old list that the Web Part was associated with.

@NewListId: The list identifier of the new list that the Web Part will be associated with.

Return Code Values: An integer which the protocol client MUST ignore.

Result Sets: MUST NOT return any result sets.

3.1.4.19 **proc_DeplGetContentTypeData**

The `proc_DeplGetContentTypeData` Stored Procedure is called during an Export Operation to get a list of Content Types. The `proc_DeplGetContentTypeData` Stored Procedure is defined using T-SQL syntax, as follows:

```
PROCEDURE proc_DeplGetContentTypeData (  
    @LastOrdinal    int  
)  
;
```

@LastOrdinal: The Export Ordinal number.

Return Code Values: An integer which the protocol client MUST ignore.

Result Sets: MUST return the following Result Set:

3.1.4.19.1 **Content Type Information Result Set**

The Content Type Information Result Set returns one row for each Content Type that will be exported. This Result Set will contain 50 rows, or one row for each content type, whichever is smaller. The Content Type Information Result Set is defined using T-SQL syntax, as follows:

```
Id                uniqueidentifier,  
ExportObjectUrl   nvarchar(260),  
Ordinal           int,  
ContentTypeId     varbinary(512),  
ParentWebId       uniqueidentifier,  
ContentTypeListId uniqueidentifier;
```

Id: This value MUST be ignored.

ExportObjectUrl: This value MUST be ignored.

Ordinal: The Export Ordinal number of this Content Type in the table of export objects.

ContentTypeId: The **content type identifier** of this Content Type.

ParentWebId: The site identifier for the parent Site of this Content Type.

ContentTypeListId: The list identifier of the list that contains this Content Type.

3.1.4.20 **proc_DeplGetContentTypesToParse**

The `proc_DeplGetContentTypesToParse` Stored Procedure is called during an Export Operation to get the list of Content Types which will be exported. The purpose of this Stored Procedure is to get a list of Content Types so that the application can look for additional, dependent objects that are necessary to correctly persist the state of these Content Types. If such objects are found, the application MUST call [proc_DeplAddExportObject](#) to add these objects to the table of export objects. `proc_DeplGetContentTypesToParse` is defined using T-SQL syntax, as follows:

```

PROCEDURE proc_DeplGetContentTypesToParse(
    @SiteId      uniqueidentifier,
    @LinksOnly   tinyint
);

```

@SiteId: The site collection identifier for the **Site Collection** containing the Content Types.

@LinksOnly: MUST be either 0 or 1. When set to 1, this Stored Procedure will only return Content Types used by link documents. When set to 0, this Stored Procedure will only return Content Types used by Documents that are not link documents.

Return Code Values: An integer which the protocol client MUST ignore.

Result Sets: MUST return the following Result Set:

3.1.4.20.1 Content Type List Result Set

The Content Type List Result Set returns a list of Content Types. This Result Set will include all of the Content Types which will be exported. Each row contains information about one Content Type. The Content Type List Result Set is defined using T-SQL syntax, as follows:

```

ContentTypeId      varbinary(512),
ContentTypeListId  uniqueidentifier,
ParentWebId        uniqueidentifier;

```

ContentTypeId: The content type identifier of the content type.

ContentTypeListId: The list identifier of the list that contains this Content Type.

ParentWebId: The site identifier of the Site that contains this list.

3.1.4.21 proc_DeplGetDocLibRootFolderUrl

The `proc_DeplGetDocLibRootFolderUrl` Stored Procedure is called to get the Store-Relative Form to the Root Folder of a document library. `proc_DeplGetDocLibRootFolderUrl` is defined using T-SQL syntax, as follows:

```

PROCEDURE proc_DeplGetDocLibRootFolderUrl(
    @SiteId      uniqueidentifier,
    @ListId      uniqueidentifier,
    @Url         nvarchar(1024) OUTPUT
);

```

@SiteId: The site collection identifier for the **Site Collection** that contains the document library.

@ListId: The list identifier for the document library whose Root Folder's URL is to be determined.

@Url: The Store-Relative Form of the document library whose list identifier is specified by `@ListId`. If such a document library does not exist, the value in `@Url` MUST NOT be changed.

Return Code Values: An integer which the protocol client MUST ignore.

Result Sets: MUST NOT return any result sets.

3.1.4.22 proc_DeplGetFileData

The proc_DeplGetFileData Stored Procedure is called to select all the available metadata information about a File to be exported, except its content stream. proc_DeplGetFileData is defined using T-SQL syntax, as follows:

```
PROCEDURE proc_DeplGetFileData(  
    @SiteId          uniqueidentifier,  
    @LastOrdinal     int,  
    @VersionSetting  int  
);
```

@SiteId: The site collection identifier for the **Site Collection** to operate on.

@LastOrdinal: All Files returned MUST have values in the Ordinal column greater than @LastOrdinal. When calling proc_DeplGetFileData for the first time, set @LastOrdinal to 0. During additional calls to retrieve more Files, set @LastOrdinal to the largest ordinal value returned in the previous call to proc_DeplGetFileData.

@VersionSetting: A value specifying which File versions to select for export which MUST be listed in the following table:

Value	Description
1	Export only the current major version of a File.
2	Export the current version of a File, which is either the most current Minor Version or the most current major version.
3	Export the most current major version and the most current Minor Version of the File.
4	Export all checked-in versions of the File.

Return Code Values: An integer which the protocol client MUST ignore.

Result Sets: MUST return seven result sets in the following order:

3.1.4.22.1 File Information Result Set

The File Information Result Set returns one row for each individual File that needs to be exported ignoring additional versions of the file. If a File has a Publishing Level of **checked out**, it MUST NOT be returned.

The File Information Result Set MUST contain 200 rows, or one row for each file, whichever is smaller. The File Information Result Set is defined using T-SQL syntax, as follows:

```
Id                uniqueidentifier,  
ExportObjectUrl  nvarchar(260),  
Ordinal          int,  
WebId            uniqueidentifier,  
ListId           uniqueidentifier,  
ParentId        uniqueidentifier,  
ScopeId         uniqueidentifier,  
DocLibRowId     int,  
MetaInfo        image,  
UIVersion       int,
```

UIVersionString	nvarchar(61),
IsGhosted	bit,
TimeCreated	datetime
TimeLastModified	datetime,
DirName	nvarchar(256),
LeafName	nvarchar(128),
SetupPath	nvarchar(255),
SetupPathUser	nvarchar(255),
SetupPathVersion	tinyint,
UnVersionedMetaInfo	image,
CheckinComment	nvarchar(1023),
AuditFlags	int,
InheritAuditFlags	int,
Level	tinyint,
HasStream	int,
Size	int,
Acl	image;

Id: The identifier of the file.

ExportObjectUrl: The full Store-Relative form URL for the file.

Ordinal: An identifier of the file that is a number used with the @LastOrdinal parameter. All rows returned MUST have an ordinal value greater than @LastOrdinal. The ordinals of all rows returned MUST be in increasing order.

WebId: The site identifier for the Site containing the file.

ListId: The list identifier of the list containing the file. If the file is not contained in a list, this value MUST be NULL.

ParentId: The identifier of the Document's parent container. The parent container could be either a folder or document library. This value MUST NOT be NULL.

ScopeId: The identifier of the scope which contains the Document.

DocLibRowId: The identifier in the document library for this item. If the file is not contained in a list, this value MUST be NULL.

MetaInfo: A **metadict** for the file. The Metadict format is specified in [\[MS-FPSE\]](#), section 2.2.2.5.3.

UIVersion: A **user interface (UI) version** number associated with the Document.

UIVersionString: A calculated column presenting the value of the user interface (UI) version column as a **displayed version** string.

IsGhosted: Indicates whether the file is **uncustomized**. The value MUST be "1" if the file is uncustomized. Otherwise, the value MUST be zero ("0").

TimeCreated: A timestamp in UTC specifying when this file was first created.

TimeLastModified: A timestamp in UTC. The value specifies when the file version returned was last saved. This MAY correspond to the actual time when the file was last modified.

DirName: The **directory name** of the file

LeafName: The **leaf name** of the file.

SetupPath: For a file that is now or once was uncustomized, 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 uncustomized.

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

SetupPathVersion: For an uncustomized document, this governs the **SetupPath** fragment starting from the template path location. The following are valid values:

Value	Description
2	The SetupPath is relative to the template location of WSSv2 on the front-end Web server.
3	The SetupPath is relative to the template location of WSSv3 on the front-end Web server.

UnVersionedMetaInfo: A Metadict for the file. The Metadict format is specified in [MS-FPSE], section 2.2.2.5.3.

CheckinComment: Comment associated with the file when this version of the file was last checked in.

AuditFlags: An **audit flags** value determining the operations to be tracked on this file.

InheritAuditFlags: An audit flags value for the operations to be tracked on this file, as determined from parent container audit flags settings.

Level: The publishing level of this version of the document.

HasStream: Identifies whether the file contents are stored in the database or on the file system. A value of 0 indicates that the file is uncustomized and stored outside of the **content database**; otherwise, this value MUST be 1.

Size: The size of the **document stream** in bytes. This value MUST not be NULL.

Acl: The binary serialization of the WSS ACL Format access control list, specified in [MS-WSSFO] section 2.2.4.6 WSS ACL Format, for this file. This is either explicitly defined or inherited from the file's parent object.

3.1.4.22.2 Latest File Versions Result Set

Latest File Versions Result Set MUST return one or two rows for each document that is valid to be exported. If a document has a Publishing Level of Checked Out, it will not be returned. If @VersionSetting is 1 then this result set MUST return one row for each document of Publishing Level of type **publish**. If @VersionSetting is not 1, the most current Publish and the current **draft** after the last **published version** is returned. If there is no document of type Publish and @VersionSetting is not 1 then only the most current Draft version of the document is returned. The Latest File Version Result Set will return information for the first 200 documents with ordinals greater than @LastOrdinal. The Latest h

Versions Information Result Set is defined using T-SQL syntax, as follows:

```
Id                uniqueidentifier,  
UIVersion         int,  
TimeCreated       datetime,
```

```

TimeLastModified      datetime,
DocFlags              int,
MetaInfo              image,
CheckinComment        nvarchar(1023)
Level                 tinyint,
SetupPath             nvarchar(255)
SetupPathUser         nvarchar(255)
SetupPathVersion      tinyint
HasStream             int
ListId                uniqueidentifier,
Size                  int;

```

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

UIVersion: A user interface (UI) version number associated with the Document.

TimeCreated: A timestamp in UTC specifying when this document was first created.

TimeLastModified: A timestamp in UTC specifying when the selected version of the document was last modified.

DocFlags: A Doc Flags value specifying information about the Document. See the [\[MS-WSSFO\]](#), Doc Flags section 2.2.2.3 for details.

MetaInfo: A Metadict for the document. The Metadict format is specified in [\[MS-FPSE\]](#), section 2.2.2.5.3.

CheckinComment: Comment associated with the document when this version of the document was last checked in.

Level: The Publishing Level of this version of the Document.

SetupPath: For a document that is now or once was uncustomized, 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 uncustomized.

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

SetupPathVersion: For an uncustomized document, this governs the **SetupPath** fragment starting from the template path location. The following are valid values:

Value	Description
2	The SetupPath is relative to the template location of WSSv2 on the front-end Web server.
3	The SetupPath is relative to the template location of WSSv3 on the front-end Web server.

HasStream: Identifies whether the Document contents are stored in the database or on the file system. A value of zero ("0") indicates that the document is uncustomized and stored outside of the content database. Otherwise, this value **MUST** be "1".

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

Size: The size of the **Document Stream** in bytes. This value MUST not be NULL.

3.1.4.22.3 Previous File Versions Result Set

The Previous File Version Result Set MUST contain 0 or more rows, for each file that remains to be exported. If @VersionSetting is not 4, then 0 rows MUST be returned. The Previous File Versions Result Set will return information for the first 200 files with ordinals greater than @LastOrdinal. The Previous File Versions Result Set is defined using T-SQL syntax, as follows:

Id	uniqueidentifier,
Version	int,
TimeCreated	datetime,
DocFlags	int,
MetaInfo	image,
CheckinComment	nvarchar(1023)
HasStream	bit,
Size	int;

Id: The identifier of the file.

Version: A user interface (UI) version number associated with the Document.

TimeCreated: A timestamp in UTC specifying when the selected version of the file was created.

DocFlags: A Doc Flags value specifying information about the file. See the [\[MS-WSSFO\]](#), Doc Flags section 2.2.2.3 for details.

MetaInfo: A Metadict for the file. The Metadict format is specified in [\[MS-FPSE\]](#), section 2.2.2.5.3.

CheckinComment: Comment associated with the file when this version of the file was last checked in.

HasStream: Identifies whether the file contents are stored in the database or on the file system. A value of zero ("0") indicates that the file is uncustomized and stored outside of the content database. Otherwise, this value MUST be "1".

Size: The size of the **Document Stream** in bytes. This value MUST not be NULL.

3.1.4.22.4 File Web Part Result Set

The File Web Part Result Set returns all the Web Part references in the files to be exported. The File Web Part Result Set will return information for the first 200 files with ordinals greater than @LastOrdinal. The File Web Part Result Set is defined using T-SQL syntax, as follows:

tp_ID	uniqueidentifier,
tp_ListID	uniqueidentifier,
tp_Type	tinyint,
tp_Flags	int,
tp_Hidden	int,
tp_Threaded	int,
tp_FPModified	int,
tp_ReadOnly	int,
tp_Scope	nvarchar(16),
tp_RecurrenceRowset	int,
tp_ModerationType	nvarchar(16),
tp_Personal	int,

tp_Ordered	int,
tp_DisplayName	nvarchar(255),
tp_ContentTypeId	varbinary(512),
tp_PageUrl	nvarchar(1024),
tp_BaseViewId	tinyint,
tp_View	ntext,
tp_Level	tinyint,
tp_IsDefault	bit,
tp_ZoneID	nvarchar(64),
tp_IsIncluded	bit,
tp_PartOrder	int,
tp_FrameState	tinyint,
tp_Version	int,
tp_WebPartTypeId	uniqueidentifier,
tp_AllUsersProperties	image,
tp_PerUserProperties	image,
tp_UserID	int,
tp_Source	ntext,
tp_CreationTime	DateTime,
DocId	uniqueidentifier,
tp_ListRootFolderUrl	nvarchar(260);

tp_ID: The identifier of the Web Part.

tp_ListID: The list identifier of the list.

tp_Type: The **page type** of this Web Part.

tp_Flags: A [\[MS-WSSFO\]](#), View Flags section 2.2.2.11 value specifying View related settings for this Web Part.

tp_Hidden: This is 1 if the view MUST NOT be displayed in enumerations of the Views of this list (for example, in a View selector user interface element). Otherwise, this is 0.

tp_Threaded: Threaded view for legacy discussion board lists (lists with a List Base Type, specified in [\[MS-WSSFO\]](#) section 2.2.3.9 List Base Type, of 3). This is 1 if client 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. Otherwise, this is 0.

tp_FPModified: This is 1 if the view has been modified by a client application such that it might not be compatible with the web interface for View modification. Otherwise, this is 0. Implementations MUST restrict modifying any properties they do not understand.

tp_ReadOnly: This is 1 if the view is read-only and implementations MUST NOT permit any modifications to its properties. Otherwise, this is 0.

tp_Scope: The [web part scope](#), section 2.2.1.2.

tp_RecurrenceRowset: This is 1 if it is required that the 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. Otherwise, this is 0.

tp_ModerationType: The [web part moderation type](#), section 2.2.1.1.

tp_Personal: This MUST be 0.

tp_Ordered: This MUST be 1 if the view displays list items based on the item order of the list (using the tp_ItemOrder column). Otherwise, this MUST be 0.

tp_DisplayName: The Display Name of the web part.

tp_ContentTypeId: A Content Type identifier value describing the valid content types of folders 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.

tp_PageUrl: The **server-relative URL** to the Document that contains this Web Part. It MUST begin with a slash. It MUST NOT end with a slash.

tp_BaseViewId: Integer referring to the View specified in the list Template for this list that was used to create this View.

tp_View: An XML string that contains the meta data of the View, if this Web Part is a View. The schema of this XML string is defined as [\[MS-WSSCAML\]](#), NoSection. If this Web Part is not a View, this MUST be NULL.

tp_Level: A publishing level value specifying the publishing status of the Web Part.

tp_IsDefault: This MUST be either 0 or 1. When set to 1, the current web part is a **default form** of the list. Otherwise, the web part is not a default form.

tp_ZoneId: The name of a Web Part Zone that the Web Part is in. This value MUST be NULL if the Web Part is not in any **Web Part zone**.

tp_IsIncluded: The Web Part Is Closed state of the Web Part. This value MUST NOT be NULL.

tp_PartOrder: Ordinal number indicating the location of the Web Part in relation to other Web Parts in the same Web Part Zone.

tp_FrameState: The Web part chrome state of the Web Part. This value MUST NOT be NULL and 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_Version: The version of the Web Part.

tp_WebPartTypeId: A value computed from the hash of the Web Part's fully qualified **assembly name**.

tp_AllUsersProperties: A serialized representation of 0 or more **customizable** properties on the Web Part. If this value is NULL, then default values will be used for all of the customizable properties on the Web Part.

tp_PerUserProperties: A serialized representation of zero or more personalizable properties on the Web Part for each user. If this value is NULL, then default values will be used for all of the personalizable properties on the Web Part. Personalizable indicates a designation for a Web Part property that can be modified for all users and for individual Users. Multiple values are maintained in the back-end database servers for this type of property: One value that is used for all users that

have not yet modified the property for themselves, and one value for each user that has modified the property for themselves.

tp_UserID: The Security principal identifier of the user who created the Web Part. This value MUST be NULL if the web part was added through a shared view.

tp_Source: The Web part properties of the Web Part in either the format specified in [\[MS-WPPS\]](#) section 2.2.4.5 or HTML format.

tp_CreationTime: A timestamp in UTC specifying when this Web Part was created.

DocId: The file that contains this Web Part.

tp_ListRootFolderUrl: This value MUST be NULL.

3.1.4.22.5 Web Part Personalization Result Set

The Web Part Personalization Result Set returns the Web Part personalizations in the files to be exported. The Web Part Personalization Result Set will return information for the first 200 files with ordinals greater than @LastOrdinal. The Web Part Personalization Result Set is defined using T-SQL syntax, as follows:

```
tp_Id                uniqueidentifier,  
tp_DocId             uniqueidentifier,  
tp_UserID            int,  
tp_PartOrder         int,  
tp_ZoneID            nvarchar(64),  
tp_IsIncluded        bit,  
tp_FrameState        tinyint,  
tp_PerUserProperties image;
```

tp_Id: The **web part identifier** of the personalized **Web Part**.

tp_DocId: The identifier of the file which contains the Web Part to be personalized. This MUST NOT be NULL.

tp_UserID: The identification number of the user who personalized the Web Part.

tp_PartOrder: Ordinal number indicating the location of the Web Part in relation to other Web Parts in the same Web Part Zone.

tp_ZoneID: The name of a Web Part Zone that the Web Part is in. This value MUST be NULL if the Web Part is not in any Web Part Zone.

tp_IsIncluded: The Web Part Is Closed State of the Web Part. This value MUST NOT be NULL.

tp_FrameState: The Web part chrome state of the Web Part. This value MUST NOT be NULL.

tp_PerUserProperties: The properties to assign to this Web Part for this user.

3.1.4.22.6 File Links Result Set

The File Links Result Set returns information about the Server-Relative URLs in the files to be exported. The File Links Result Set will return information for the first 200 list items with ordinals greater than @LastOrdinal. The File Links Result Set is defined using T-SQL syntax, as follows:

Id	uniqueidentifier,
{TargetId}	uniqueidentifier,
TargetUrl	nvarchar(261),
WebPartId	uniqueidentifier,
LinkNumber	int,
Type	tinyint,
Security	tinyint,
Dynamic	tinyint,
ServerRel	bit,
Level	tinyint,
Search	ntext;

Id: The document identifier of the document.

{TargetId}: The document identifier of the item the links point to. The value MUST not be NULL.

TargetUrl: The server-relative URL of the target document.

WebPartId: The identifier of the Web Part containing the link.

LinkNumber: An ordinal value denoting the relative order of this **hyperlink** within the source of the Document, Web Part, or Field being processed.

Type: Type of the link. Values specified in [\[MS-WSSDLIM\]](#) section 3.1.4.8 proc_AL.

Security: A one byte (tinyint) value represented as a single upper case ASCII character specifying the Link's type of URL. Values specified in [\[MS-WSSDLIM\]](#) section 3.1.4.8 proc_AL.

Dynamic: A one byte (tinyint) value represented as a single upper case ASCII character which tracks various special Link Types. Values specified in [\[MS-WSSDLIM\]](#) section 3.1.4.8 proc_AL.

ServerRel: This value MUST be 1.

Level: The Publishing Level of this version of the Document.

Search: Search parameters for GLOSSARY **forward links**. The link source starting at either the query string signifier '?' or the bookmark signifier '#'.

3.1.4.22.7 File Event Receivers Result Set

The Event Receivers Result Set contains information about the **event receivers** defined for the files in the given Site Collection. The Event Receivers Result Set MUST return zero rows. [<3>](#)

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

Id	uniqueidentifier,
HostId	uniqueidentifier,
WebId	uniqueidentifier,
Name	nvarchar(256),
HostType	int,
ItemId	int,
Type	int,
SequenceNumber	int,
Assembly	nvarchar(256),
Class	nvarchar(256),
Data	nvarchar(256),
Filter	nvarchar(256),

```
Credential          int;
```

Id: The identifier of the event receiver.

HostId: The identifier of the object on which the event receiver is registered.

WebId: The site identifier for a Site that contains the event receiver.

Name: The display name of the event receiver.

HostType: The identifier of the **event host** type on which the event receiver is registered.

ItemId: The identifier of the list item on which the event receiver is registered.

Type: The identifier of the event receiver type of the event tracked by the event receiver.

SequenceNumber: An ordinal that determines the order in which the event receiver is triggered.

Assembly: The name of the .NET **assembly** that contains the class definition of the event receiver.

Class: The name of the .NET class definition for the Event receiver.

Data: **Unicode** string data specific to the implementation of the Event receiver.

Filter: Unused and MUST be ignored.

Credential: Unused and MUST be ignored.

3.1.4.23 proc_DeplGetFolderById

The `proc_DeplGetFolderById` stored procedure returns the site identifier and server-relative URL for a Folder specified by the **folder identifier**. `proc_DeplGetFolderById` is defined using T-SQL syntax, as follows:

```
PROCEDURE proc_DeplGetFolderById(  
    @SiteId          uniqueidentifier,  
    @FolderId        uniqueidentifier,  
    @DeleteRecycleBinItems bit= 0  
);
```

@SiteId: The site collection identifier for the **Site Collection** that contains the Folder specified by @FolderID.

@FolderId: The folder identifier for the Folder in the Site Collection specified by @SiteId.

@DeleteRecycleBinItems: If this is 1 and the Folder specified by @FolderId is in the **Recycle Bin**, then the Folder and all items deleted in the same transaction MUST be emptied from the Recycle Bin. If this is not 1 and the Folder is in the Recycle Bin, then error 30001 MUST be returned.

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

Value	Description
0	Execution finished. A Result Set MUST be returned. This MUST also be the value returned if the @SiteId or @FolderId do not match an existing item.

Value	Description
80	The specified Folder exists, but belongs to a different Site Collection.
87	@SiteId or @FolderId is NULL.
30001	The specified Folder is in the Recycle Bin and @DeleteRecycleBinItems is not 1. The specified Folder is in the Recycle Bin and @DeleteRecycleBinItems is 1, but the attempt to empty the Recycle Bin failed.

Result Sets: MUST return the following Result Set:

3.1.4.23.1 Folder Location Result Set

The Folder Location Result Set returns information about the specified Folder. On successful execution, the Folder Location Result Set MUST be returned and MUST contain 0 or one rows. One row MUST be returned if @SiteId and @FolderId correspond to an existing item. 0 rows MUST be returned in all other cases.

The Folder Location Result Set is defined using T-SQL syntax, as follows:

```
WebId          uniqueidentifier,
Url            nvarchar(261);
```

WebId: The site identifier for the Site containing the specified Folder.

Url: The server-relative URL of the Folder. It MUST begin with a slash.

3.1.4.24 proc_DeplGetFolderData

The proc_DeplGetFolderData stored procedure returns information about the Folders to be processed during an Export Operation. proc_DeplGetFolderData is defined using T-SQL syntax, as follows:

```
PROCEDURE proc_DeplGetFolderData (
    @SiteId          uniqueidentifier,
    @LastOrdinal     int,
    @VersionSetting  int
);
```

@SiteId: The site collection identifier for the **Site Collection** to operate on.

@LastOrdinal: All Folders returned by proc_DeplGetFolderData MUST have values in the Ordinal column of the [Folder Information result set](#), specified in section 3.1.4.23.1, greater than @LastOrdinal. When calling proc_DeplGetFolderData the first time, set @LastOrdinal to 0. When calling proc_DeplGetFolderData to retrieve more Folders, set @LastOrdinal to the largest Ordinal value returned in the previous call to proc_DeplGetFolderData.

@VersionSetting: If this is 1, proc_DeplGetFolderData MUST only return information about Folders with a Publishing Level of published. If this is not 1, proc_DeplGetFolderData MUST return exactly one row for each unique Folder. If a version of a folder has a Publishing Level of Checked Out, information about a different version MUST be returned. If the folder only has information with a publishing level of checked out, proc_DeplGetFolderData MUST NOT return any information about that folder.

Return Code Values: An integer which the protocol client MUST ignore.

Result Sets: MUST return the following Result Set:

3.1.4.24.1 Folder Information Result Set

The Folder Information Result Set returns one row for each Folder that needs to be exported. If a Folder only has a version with a Publishing Level of Checked Out, it will not be returned. If @SiteId is not found or there are no Folders that need to be exported, the Result Set MUST be empty.

The Folder Information Result Set MUST contain 200 rows, or one row for each folder that remains to be exported, whichever is smaller. The Folder Information Result Set is defined using T-SQL syntax, as follows:

Id	uniqueidentifier,
ExportObjectUrl	nvarchar(260),
Ordinal	int,
ParentWebId	uniqueidentifier,
LeafName	nvarchar(128),
ListId	uniqueidentifier,
DocLibRowId	int,
WelcomePageUrl	nvarchar(260),
WelcomePageParameters	ntext,
IsWebRootFolder	bit,
Acl	image,
MetaInfo	image,
TimeCreated	datetime,
TimeLastModified	datetime;

Id: The identifier of the Folder.

ExportObjectUrl: The full Store-Relative Form URL for the Folder.

Ordinal: An identifier of the Folder that is a number used with the @LastOrdinal parameter. All rows returned MUST have an ordinal value greater than @LastOrdinal. The ordinals of all rows returned MUST be in increasing order.

ParentWebId: The site identifier for the Site containing the Folder.

LeafName: The Leaf Name of the Folder.

ListId: The list identifier of the list containing the Folder. If the Folder is not contained in a list, this value MUST be NULL.

DocLibRowId: The identifier in the document library for this item. If the Folder is not contained in a list, this value MUST be NULL.

WelcomePageUrl: An optional URL 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 the Folder. If a welcome Page is not specified, this value MUST be NULL.

WelcomePageParameters: Optional URL parameters to use with the WelcomePageUrl value. If this value is not NULL, it MUST contain a query string starting with "?", a fragment parameter starting with "#", or both a query string and a fragment parameter, starting with "?" and containing "#". If the parameters are not specified, this value MUST be NULL.

IsWebRootFolder: If the Folder is the root of a Site, this MUST be 1. Otherwise, this MUST be 0.

Acl: The binary serialization of the WSS ACL Format access control list, specified in [\[MS-WSSFO\]](#) section 2.2.4.6 WSS ACL Format, for this Folder. This is either explicitly defined or inherited from the Folder's parent object.

MetaInfo: A **Metadict** for the Folder. The Metadict format is specified in [\[MS-FPSE\]](#), section 2.2.2.5.3.

TimeCreated: Date/time value in UTC specifying when this Folder was created.

TimeLastModified: Date/time value in UTC specifying when the Folder was last saved. This value does not necessarily correspond to the actual time when the Folder was last modified.

3.1.4.25 **proc_DeplGetIncrementalChanges**

The `proc_DeplGetIncrementalChanges` stored procedure is called to populate the temporary table of export objects based on incremental changes in the Change Log. `proc_DeplGetIncrementalChanges` is defined using T-SQL syntax, as follows:

```
PROCEDURE proc_DeplGetIncrementalChanges(  
    @SiteId                uniqueidentifier,  
    @SearchEntireSite      tinyint,  
    @ChangeTimeStart       datetime,  
    @ChangeNumberStart    int,  
    @ChangeTimeEnd        datetime,  
    @ChangeNumberEnd      int,  
    @IncludeUsers         tinyint,  
    @IncludeGroups        tinyint,  
    @IsMajorFileOnly      tinyint,  
    @MinChangeTime        datetime OUTPUT,  
    @MinChangeNumber      int OUTPUT  
);
```

@SiteId: The site collection identifier for the **Site Collection** to operate on.

@SearchEntireSite: If `@SearchEntireSite` is 1, `proc_DeplGetIncrementalChanges` MUST search the entire site collection for incremental changes. If this is not 1, `proc_DeplGetIncrementalChanges` MUST be limited to the Sites that are included in the Incremental Search Scope Table.

@ChangeTimeStart: All changes added to the table of export objects MUST have occurred at or after `@ChangeTimeStart`. `@ChangeTimeStart` MUST be ignored if `@SearchEntireSite` is not 1.

@ChangeNumberStart: All changes added to the table of export objects MUST have an identifier greater than `@ChangeNumberStart`. `@ChangeNumberStart` MUST be ignored if `@SearchEntireSite` is not 1.

@ChangeTimeEnd: All changes added to the table of export objects MUST have occurred at or before `@ChangeTimeEnd`.

@ChangeNumberEnd: All changes added to the table of export objects MUST have an identifier less than or equal to `@ChangeNumberEnd`.

@IncludeUsers: This value is not used and MUST be ignored.

@IncludeGroups: This value is not used and MUST be ignored.

@IsMajorFileOnly: If this is 1, only objects with a Publishing Level of published will be added. If this is 0, exactly one row for each unique object will be added, except information about versions with a Publishing Level of Checked Out MUST NOT be returned.

@MinChangeTime: When proc_DeplGetIncrementalChanges returns, this MUST be set to the time of the earliest change in the Change Log.

@MinChangeNumber: When proc_DeplGetIncrementalChanges returns, this MUST be set to the Change log identifier with the lowest value in the Change Log.

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

Value	Description
0	Execution finished.
30000	This MUST be returned if there are no changes in the Change Log. This MUST be returned if the stored procedure did not find a valid entry in the change log that matches either @ChangeTimeStart or @ChangeNumberStart. If @SearchEntireSite is 1, this value MUST be returned if @ChangeTimeStart or @ChangeNumberStart is NULL.

Result Sets: MUST NOT return any result sets.

3.1.4.26 proc_DeplGetListData

The proc_DeplGetListData stored procedure is called to return information about the lists to be processed during an export operation. proc_DeplGetListData is defined using T-SQL syntax, as follows:

```

PROCEDURE proc_DeplGetListData (
    @LastOrdinal          int,
    @SiteId               uniqueidentifier,
    @IsIncremental        bit,
    @ChangeTimeStart      datetime= NULL,
    @ChangeNumberStart    int= NULL,
    @ChangeTimeEnd        datetime= NULL,
    @ChangeNumberEnd      int= NULL
);

```

@LastOrdinal: The information for all lists returned by proc_DeplGetListData MUST have values in the Ordinal column greater than @LastOrdinal. When calling proc_DeplGetListData the first time, set @LastOrdinal to 0. When calling proc_DeplGetListData to retrieve more lists, set @LastOrdinal to the largest Ordinal value returned in the previous call to proc_DeplGetListData.

@SiteId: The site collection identifier for the **Site Collection** to operate on.

@IsIncremental: If this is 1, the [Deleted Fields result set](#), specified in section 3.1.4.25.4, will be returned. Otherwise, the [Null List result set](#), specified in section 3.1.4.25.5, will be returned.

@ChangeTimeStart: @ChangeTimeStart is only used if @IsIncremental is 1. If this is not NULL, the entire Site Collection will be searched for incremental changes that result in deleted Fields. All changes returned MUST have occurred at or after @ChangeTimeStart. If this is NULL, fields returned will be limited to the Sites that are included in the Incremental Search Scope Table.

@ChangeNumberStart: @ChangeNumberStart is only used if @IsIncremental is 1 and @ChangeNumberStart is not NULL. All Field deletion changes returned MUST have a Change log identifier greater than @ChangeNumberStart.

@ChangeTimeEnd: @ChangeTimeEnd is only used if @IsIncremental is 1. All Field deletion changes returned MUST have occurred at or before @ChangeTimeEnd.

@ChangeNumberEnd: @ChangeNumberEnd is only used if @IsIncremental is 1. All Field deletion changes returned MUST have a Change log identifier less than or equal to @ChangeNumberEnd.

Return Code Values: An integer which the protocol client MUST ignore.

Result Sets: MUST return four result sets in the following order:

3.1.4.26.1 List Information Result Set

This Result Set MUST be returned.

The List Information Result Set MUST contain 50 rows, or one row for each list that remains to be exported, whichever is smaller. The List Information Result Set is defined using T-SQL syntax, as follows:

Id	uniqueidentifier,
ExportObjectUrl	nvarchar(260),
Ordinal	int,
tp_Title	nvarchar(255),
tp_Description	ntext,
tp_Modified	datetime,
tp_Created	datetime,
tp_Version	int,
tp_BaseType	int,
tp_FeatureId	uniqueidentifier,
tp_ServerTemplate	int,
tp_RootFolder	uniqueidentifier,
tp_Template	uniqueidentifier,
tp_ReadSecurity	int,
tp_WriteSecurity	int,
tp_Fields	ntext,
tp_Direction	int,
tp_Flags	bigint,
tp_ThumbnailSize	int,
tp_WebImageWidth	int,
tp_WebImageHeight	int,
tp_ImageUrl	nvarchar(255),
tp_Author	int,
tp_EventSinkAssembly	nvarchar(255),
tp_EventSinkClass	nvarchar(255),
tp_EventSinkData	nvarchar(255),
tp_EmailInsertsFolder	nvarchar(255),
tp_EmailAlias	nvarchar(128),
tp_DocTemplateUrl	nvarchar(261),
tp_ContentTypes	ntext,
Acl	image,
tp_MaxMajorVersionCount	int,
tp_MaxMajorwithMinorVersionCount	int,
tp_SendToLocation	nvarchar(512);

Id: The identifier of the list.

ExportObjectUrl: The full Store-Relative Form URL for the list.

Ordinal: An identifier of the list that is a number used with the @LastOrdinal parameter. All rows returned MUST have an ordinal value greater than @LastOrdinal. The ordinals of all rows returned MUST be in increasing order.

tp_Title: The title of this list for display in the user interface.

tp_Description: The description of this list for display in the user interface.

tp_Modified: A timestamp in UTC specifying when this list was last modified.

tp_Created: A timestamp in UTC specifying when this list was created.

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

tp_BaseType: This specifies the List Base Type, specified in [\[MS-WSSFO\]](#) section 2.2.3.9 List Base Type, of this list.

tp_FeatureId: The identifier for the feature that defines the base schema of this list.

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

tp_RootFolder: The identifier of the root Folder of this list.

tp_Template: The identifier of the default template to use when a Document is created in this document library. tp_Template MUST be NULL if this list is not a document library.

tp_ReadSecurity: This signifies special restrictions that MUST be placed on list item access. It MUST be one of the following values:

Value	Description
1	No special restrictions.
2	Users MUST see only their own list items. The front-end Web server MUST NOT display list items to users without the ManageLists right unless the list item was created by that user. The ManageLists right is specified in [MS-WSSFO] section 2.2.2.13 WSS Rights Mask.

tp_WriteSecurity: This signifies special restrictions that can be placed on list item update. It MUST be one of the following values:

Value	Description
1	No special restrictions.
2	Users can change only their own list items. The front-end Web server MUST NOT allow users without the ManageLists right to update a list item unless the list item was created by that user.
4	Users will not update any list items in this list. The front-end Web server MUST NOT allow users without the ManageLists right to add or update list items in this list.

tp_Fields: This value is an XML string that specifies the Fields used by this list. The schema of the XML string is defined by [\[MS-WSSCAML\]](#) section 2.4.1.13 FieldDefinitionDatabaseWithVersion.

tp_Direction: An enumerated value specifying the direction of text flow for user interface elements presented by this list. It MUST be one of the following values:

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

tp_Flags: A list flags value describing this list.

tp_ThumbnailSize: The width, in pixels, that MUST be used when creating thumbnail images of list items within this list. If the list is not a **picture library**, this MUST be NULL.

tp_WebImageWidth: The width, in pixels, that MUST be used when displaying images of list items within this list. If the list is not a picture library, this MUST be NULL.

tp_WebImageHeight: The height, in pixels, that MUST be used when displaying images of list items within this list. If the list is not a picture library, this MUST be NULL.

tp_ImageUrl: The server-relative URL of the image used to represent this list.

tp_Author: The identifier in the UserInfo Table, specified in [\[MS-WSSFO\]](#) section 2.2.7.10 UserInfo Table, of the user who is listed as creating this list.

tp_EventSinkAssembly: The assembly name of the implementation of the backwards compatible event receiver associated with this list.

tp_EventSinkClass: The **fully qualified class name** of the implementation of the backwards compatible event receiver associated with this list.

tp_EventSinkData: Additional data persisted on behalf of the backwards compatible event receiver implementation to be passed to the backwards compatible event receiver associated with this list.

tp_EmailInsertsFolder: This MUST be ignored.

tp_EmailAlias: The e-mail alias of the list. This alias is used to allow files to be sent directly to this list through an implementation-specific e-mail handling feature.

tp_DocTemplateUrl: The Server-relative URL of the tp_Template Document. It MUST begin with a slash.

tp_ContentTypes: **XML** data specifying the Content Types registered for this list. The XML Schemas for a Content Type are defined in wss.xsd as ContentTypeDefinition.

Acl: The binary serialization of the WSS ACL Format access control list, specified in [\[MS-WSSFO\]](#) section 2.2.4.6 WSS ACL Format, for this list. This is either explicitly defined or inherited from the parent object of the list.

tp_MaxMajorVersionCount: If the list has versioning enabled, this field 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 NULL specifies that versions MUST not automatically be removed for this list.

tp_MaxMajorwithMinorVersionCount: If the list has versioning enabled, this field contains the number of major versions that will have their associated minor versions retained for this document. All minor versions associated with a major version that is more than tp_MaxMajorwithMinorVersionCount removed from the current major version of the document are automatically removed at version creation time. A value of NULL specifies that versions MUST not automatically be removed for this list.

tp_SendToLocation: The title and URL for the "Send To Location" configured on this list. The **SendToLocation** is an implementation specific feature that allows users to manually save copies of list items and Documents to the remote location.

3.1.4.26.2 List Web Parts Result Set

This Result Set MUST be returned.

The List Web Parts Result Set contains information about the Web Parts defined for the Lists to be exported. The List Web Parts Result Set MUST contain one row for each Web Part associated with the first 50 lists to be exported with export ordinals greater than @LastOrdinal, or if there are less than 50 lists with export ordinals greater than @LastOrdinal, all of them MUST be returned. This Result Set is ordered by the Web Part creation time.

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

```

tp_ID                uniqueidentifier,
tp_ListID            uniqueidentifier,
tp_Type              tinyint,
tp_Flags              int,
tp_Hidden             bit,
tp_Threaded          bit,
tp_FPModified        bit,
tp_ReadOnly          bit,
tp_Scope             nvarchar(256),
tp_RecurrenceRowset bit,
tp_ModerationType   nvarchar(256),
tp_Personal          bit,
tp_Ordered           bit,
tp_DisplayName       nvarchar(255),
tp_ContentTypeId     varbinary(512),
tp_PageUrl           nvarchar(261),
tp_BaseViewId        tinyint,
tp_View              ntext,
tp_Level             tinyint,
tp_IsDefault         bit,
tp_ZoneID            nvarchar(64),
tp_IsIncluded        bit,
tp_PartOrder         int,
tp_FrameState        tinyint,
Id                   uniqueidentifier,
OnQuickLaunch        bit,
tp_MobileView        bit,
tp_MobileDefaultView bit,
tp_ContentTypeIdDefault bit,
tp_LockWeb           bit,
tp_FailIfEmpty       bit,
tp_FreeForm          bit,
tp_FileDialog        bit,
tp_AggregationView  bit,
tp_IncludeRootFolder bit,

```


tp_IncludeVersions bit;

tp_ID: The identifier of the Web Part.

tp_ListID: The list identifier of the list.

tp_Type: The page type of this Web Part.

tp_Flags: A [\[MS-WSSFO\]](#) section 2.2.2.11 View Flags value specifying View related settings for this Web Part.

tp_Hidden: This is 1 if the view MUST NOT be displayed in enumerations of the Views of this list (for example, in a View selector user interface element). Otherwise, this is 0.

tp_Threaded: Threaded view for legacy discussion board lists(lists with a List Base Type, specified in [\[MS-WSSFO\]](#) section 2.2.3.9 List Base Type, of 3). This is 1 if client implementations SHOULD display results in a threaded fashion, and paging of results SHOULD be done in terms of threads instead of by individual list items. Otherwise, this is 0.

tp_FPModified: This is 1 if the View has been modified by a client application such that it might not be compatible with the browser-based interface for View modification. Otherwise, this is 0. Implementations MUST restrict modifying any properties they do not understand.

tp_ReadOnly: This is 1 if the view is read-only and implementations MUST not allow any modifications to its properties. Otherwise, this is 0.

tp_Scope: The [web part scope](#), specified in section 2.2.1.2.

tp_RecurrenceRowset: This MUST be 1 if it is required that the View's data be expanded based on a recurring calendar event. For example, having a View of list **item** data for the first Thursday of every month. Otherwise, this MUST be 0.

tp_ModerationType: The [web part moderation type](#), specified in section 2.2.1.1.

tp_Personal: This MUST be 0.

tp_Ordered: This MUST be 1 if the view displays list items based on the item order of the list (using the tp_ItemOrder column). Otherwise, this MUST be 0.

tp_DisplayName: The Display Name, for use in the user interface, defined for this list.

tp_ContentTypeId: A Content Type identifier value specifying the valid folder content type 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.

tp_PageUrl: The server-relative URL to the Document that contains this Web Part. It MUST begin with a slash. It MUST NOT end with a slash.

tp_BaseViewId: The **base view identifier** for the Web Part.

tp_View: A value containing implementation-specific XML used when processing this Web Part. If this Web Part is not a View, this MUST be NULL.

tp_Level: A Publishing Level value specifying the publishing status of the Web Part.

tp_IsDefault: This is 1 if the view is to be used as the default form for the corresponding list. Otherwise, this is 0.

tp_ZoneId: The name of a Web Part Zone.

tp_IsIncluded: The Web Part Is Closed State of the Web Part. This value MUST NOT be NULL.

tp_PartOrder: Ordinal number indicating the location of the Web Part in relation to other Web Parts in the same Web Part Zone.

tp_FrameState: The Web part chrome state of the Web Part. This value MUST NOT be NULL and 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.

Id: The identifier of the Document that contains this Web Part.

OnQuickLaunch: This MUST be 1 if the document is in the Site's Navigation Structure. Otherwise, this MUST be 0.

tp_MobileView: This MUST be 1 if the view is intended for display on **mobile devices**. Otherwise, this MUST be 0.

tp_MobileDefaultView: This MUST be 1 if the view MUST be displayed as the default View of this list when a mobile View is requested. Otherwise, this MUST be 0.

tp_ContentTypeDefault: This MUST be 1 if the view MUST be the default view presented when a View is requested for a particular **Content Type**. Otherwise, this MUST be 0.

tp_LockWeb: Value is unused and MUST be ignored.

tp_FailIfEmpty: This MUST be 1 when the query for this View returns no rows, implementations of the Front-End Web Server MUST return an HTTP 410 error when displaying this View as part of an HTTP request, instead of displaying a normal empty View body. Otherwise, this MUST be 0.

tp_FreeForm: This MUST be 1 if data is presented in a nontabular fashion. Implementations SHOULD format results in a manner compatible with free-form presentation. Otherwise, this MUST be 0.

tp_FileDialog: This MUST be 1 if the View is suitable for displaying in an HTML-based file navigation dialog to client applications. Otherwise, this MUST be 0.

tp_AggregationView: This MUST be 1 if the View has functionality for aggregating data across multiple **XML documents** within an XML form library. Otherwise, this MUST be 0.

tp_IncludeRootFolder: This MUST be 1 if the View of a Folder MUST fetch the list item for the Folder of the view, in addition to the standard behavior of fetching all list items contained within it. Otherwise, this MUST be 0.

tp_IncludeVersions: This MUST be 1 if the View displays historical versions of list items. Otherwise, this MUST be 0.

3.1.4.26.3 Event Receivers Result Set

This Result Set MUST be returned.

The Event Receivers Result Set contains information about the event receivers defined for the lists to be exported. The Event Receivers Result Set MUST contain one row for each event receiver. The event receivers in the Result Set MUST be registered with the first 50 lists to be exported with export ordinals greater than @LastOrdinal, or if there are less than 50 lists with export ordinals greater than @LastOrdinal, information about all of their event receivers MUST be returned. If @SiteId does not match an existing Site Collection, this Result Set MUST be empty.

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

```
Id                uniqueidentifier,
HostId            uniqueidentifier,
WebId             uniqueidentifier,
Name              nvarchar(256),
HostType          int,
ItemId           int,
Type             int,
SequenceNumber   int,
Assembly          nvarchar(256),
Class            nvarchar(256),
Data             nvarchar(256),
Filter           nvarchar(256),
Credential       int;
```

Id: The identifier of the event receiver.

HostId: The identifier for the list on which the event receiver is registered.

WebId: The site identifier for the Site that contains the event receiver.

Name: The display name of the event receiver.

HostType: The identifier of the event host type, specified in [\[MS-WSSFO\]](#) section 2.2.3.5 event host type, on which the event receiver is registered.

ItemId: This MUST be NULL.

Type: The identifier of the event receiver type, specified in [\[MS-WSSFO\]](#) section 2.2.3.6 event receiver type, of the event tracked by the event receiver.

SequenceNumber: An ordinal that determines the order in which the event receiver is triggered.

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

Class: The name of the .NET class definition for the event receiver.

Data: Unicode string data specific to the implementation of the event receiver.

Filter: Unused and MUST be ignored.

Credential: Unused and MUST be ignored.

3.1.4.26.4 Deleted Fields Result Set

This Result Set MUST ONLY be returned if @IsIncremental is 1.

The Deleted Fields Result Set returns information about the Fields marked as deleted from the lists to be exported. The Deleted Fields Result Set will return the deleted Fields associated with the first

50 lists to be exported with export ordinals greater than @LastOrdinal, or if there are less than 50 lists with export ordinals greater than @LastOrdinal, all of them MUST be returned.

The Deleted Fields Result Set is defined using T-SQL syntax, as follows:

```
ListId          uniqueidentifier,  
FieldId        uniqueidentifier;
```

ListId: The list identifier of the list that contained the deleted Field.

FieldId: The Field identifier of the deleted Field.

3.1.4.26.5 Null List Result Set

This Result Set MUST ONLY be returned if @IsIncremental is not 1.

The Null List Result Set is a placeholder that MUST contain a single row. The Null List Result Set is defined using T-SQL syntax, as follows:

```
ListId          uniqueidentifier;
```

ListId: This MUST have a value of NULL.

3.1.4.27 proc_DeplGetListItemAndReserveId

The proc_DeplGetListItemAndReserveId stored procedure is called to determine if a Document already exists in a list, and if not, reserve an identifier for adding a new Document. proc_DeplGetListItemAndReserveId is defined using T-SQL syntax, as follows:

```
PROCEDURE proc_DeplGetListItemAndReserveId(  
    @SiteId          uniqueidentifier,  
    @ListId          uniqueidentifier,  
    @DocId           uniqueidentifier OUTPUT,  
    @IsDeleted       bit OUTPUT,  
    @ListItemId      int OUTPUT,  
    @WebId           uniqueidentifier,  
    @OriginalListItemId int,  
    @IsIdTaken       bit OUTPUT,  
    @ListItemUrl     nvarchar(260) OUTPUT,  
    @DeleteRecycleBinItems bit= 0  
);
```

@SiteId: The site collection identifier for the **Site Collection** to operate on.

@ListId: The list identifier for the list to operate on.

@DocId: The document identifier for the document. Either @DocId or @ListItemUrl MUST have a valid input value to get a valid output for @DocId and @ListItemUrl. If the input value in @DocId does not match the identifier of an existing Document, the output value of @DocId MUST be the identifier of the Document referred to by the input value of @ListItemUrl. Otherwise, the output value of @DocId MUST be the same as the input value of @DocId.

@IsDeleted: When proc_DeplGetListItemAndReserveId returns, this MUST output 1 if the Document is marked for deletion in the Recycle Bin. Otherwise, the output MUST be 0.

@ListItemId: If the Document exists, `proc_DeplGetListItemAndReserveId` MUST output in `@ListItemId` the list item identifier of the row in the document library which represents this Document. If the Document does not exist, `proc_DeplGetListItemAndReserveId` MUST output the next available identifier reserved by `proc_DeplGetListItemAndReserveId`.

@WebId: The site identifier for the Site to operate on.

@OriginalListItemId: If the list item does not exist and `@OriginalListItemId` is not already used as a list item identifier in this list, then `proc_DeplGetListItemAndReserveId` MUST reserve `@OriginalListItemId` as a list item identifier and the output of `@ListItemId` MUST be `@OriginalListItemId`. If the list item does exist or if `@OriginalListItemId` is already in use, this MUST be ignored.

@IsIdTaken: If the identifier specified in `@OriginalListItemId` is already being used as the identifier of a list item, output 1. Otherwise, output 0.

@ListItemUrl: Either `@DocId` or `@ListItemUrl` MUST have a valid input value to get a valid output for `@DocId` and `@ListItemUrl`. The input value of `@ListItemUrl` MUST be a Store-Relative form URL. If the Document matching `@DocId` exists, then `proc_DeplGetListItemAndReserveId` MUST output the Server-Relative URL of the Document in `@ListItemUrl`. Otherwise, the output value of `@ListItemUrl` MUST be the same as its input value.

@DeleteRecycleBinItems: If this is 1, the Document exists, and the Document is marked for deletion, the Document and all items deleted in the same transaction MUST be emptied from the Recycle Bin. If the Document is removed from the Recycle Bin, `@IsDeleted` MUST have output 0. If `@DeleteRecycleBinItems` is 0, the Recycle Bin MUST NOT be emptied.

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

Value	Description
0	The document does not exist or is in the Recycle Bin.
1	The document already exists.
87	<code>@SiteId</code> is NULL or <code>@ListId</code> is NULL.

Result Sets: MUST NOT return any result sets.

3.1.4.28 `proc_DeplGetListItemByDocId`

The `proc_DeplGetListItemByDocId` Stored Procedure is called to retrieve information about a Document in a list. `proc_DeplGetListItemByDocId` is defined using T-SQL syntax, as follows:

```
PROCEDURE proc_DeplGetListItemByDocId(  
    @SiteId          uniqueidentifier,  
    @ListId          uniqueidentifier,  
    @DocId           uniqueidentifier,  
    @IsDeleted       bit OUTPUT,  
    @ListItemId      int OUTPUT  
);
```

@SiteId: The site collection identifier for the **Site Collection** containing this Document.

@ListId: The list identifier for the list containing this Document.

@DocId: The document identifier of the Document.

@IsDeleted: Outputs 1 if the Document is marked for deletion. Otherwise, outputs 0.

@ListItemId: Outputs the list item identifier of the row in the Document library. If @SiteId, @ListId, or @DocId do not match any existing objects, the output value of @ListItemId MUST be the same as its input value.

Return Code Values: An integer which the protocol client MUST ignore.

Result Sets: MUST NOT return any result sets.

3.1.4.29 **proc_DeplGetListItemByUrl**

The `proc_DeplGetListItemByUrl` Stored Procedure is called to retrieve information about a list item based on its Directory Name and Leaf Name. `proc_DeplGetListItemByUrl` is defined using T-SQL syntax, as follows:

```
PROCEDURE proc_DeplGetListItemByUrl (
    @SiteId          uniqueidentifier,
    @ListId          uniqueidentifier,
    @DirName        nvarchar(256),
    @LeafName       nvarchar(128),
    @IsDeleted      bit OUTPUT,
    @ListItemId     int OUTPUT
);
```

@SiteId: The site collection identifier for the **Site Collection** containing this list.

@ListId: The list identifier for the list containing this list item.

@DirName: The Directory Name of the list item.

@LeafName: The Leaf Name of the list item.

@IsDeleted: Outputs 1 if the list item is marked for deletion. Otherwise, outputs 0.

@ListItemId: The list item identifier of the row in the list which represents this list item. If the list item is not found, the output value of @ListItemId MUST be the same as its input value.

Return Code Values: An integer which the protocol client MUST ignore.

Result Sets: MUST NOT return any result sets.

3.1.4.30 **proc_DeplGetListItemData**

The `proc_DeplGetListItemData` stored procedure is called to return information about the list items to be processed during an export operation. `proc_DeplGetListItemData` is defined using T-SQL syntax, as follows:

```
PROCEDURE proc_DeplGetListItemData (
    @SiteId          uniqueidentifier,
    @LastOrdinal    int,
    @VersionSetting int,
    @ListId         uniqueidentifier,
    @Columns       nvarchar(1700)
);
```

);

@SiteId: The site collection identifier for the **Site Collection** containing the list items.

@LastOrdinal: All list items returned MUST have export ordinals greater than @LastOrdinal. When calling proc_DeplGetListItemData for the first time, set @LastOrdinal to 0. When calling proc_DeplGetListItemData to retrieve more data, set @LastOrdinal to the largest Ordinal value returned in the previous call.

@VersionSetting: If this is 1, proc_DeplGetListItemData MUST return only current information with a Publishing Level of Published. If this is 4, proc_DeplGetListItemData MUST return data for all versions, except data about versions with a Publishing Level of Checked Out. For all other values, proc_DeplGetListItemData MUST return current version data, except data about versions with a Publishing Level of Checked Out.

@ListId: The list identifier for the list containing the list items.

@Columns: A comma-separated list of application-defined fields in the AllUserData Table, specified in [\[MS-WSSFO\]](#) section 2.2.7.3 AllUserData Table, to return in the Result Set. These are the columns marked as nvarchar#, ntext#, sql_variant#, int#, float#, datetime#, bit#, and uniqueidentifier1 in the description of AllUserData Table. Example: 'ntext1, ntext3, int10'.

Return Code Values: An integer which the protocol client MUST ignore.

Result Sets: MUST return nine result sets in the following order:

3.1.4.30.1 List Item Data Result Set

The List Item Data Result Set returns information about the current version of list items to be exported. The List Item Data Result Set MUST contain one row for each of the first 200 list items with export ordinals greater than @LastOrdinal, or one row for each list item that remains to be exported, whichever is smaller. Information about versions with a Publishing Level of Checked Out MUST NOT be returned.

The List Item Data Result Set is defined using T-SQL syntax, as follows:

Id	uniqueidentifier,
ExportObjectUrl	nvarchar(260),
Ordinal	int,
ParentId	uniqueidentifier,
DirName	nvarchar(256),
LeafName	nvarchar(128),
IntId	int,
tp_RowOrdinal	int,
tp_UIVersion	int,
tp_Author	int,
tp_Editor	int,
tp_Modified	datetime,
tp_Created	datetime,
tp_Ordering	varchar(512),
tp_HasAttachment	bit,
tp_ModerationStatus	int,
tp_IsCurrent	bit,
tp_ItemOrder	float,
tp_InstanceID	int,
tp_CopySource	nvarchar(260),

tp_HasCopyDestinations	bit,
tp_AuditFlags	int,
tp_InheritAuditFlags	int,
tp_ContentType	nvarchar (255),
tp_ContentTypeId	varbinary (512),
tp_ThreadIndex	varbinary (512),
tp_UIVersionString	nvarchar (61),
Id	uniqueidentifier,
Type	tinyint,
FolderId	uniqueidentifier,
Acl	image,
UserLoginName	nvarchar (255),
GroupName	nvarchar (255),
tp_Level	tinyint;

Id: Identifier of the list item 'that is used by the export process.

ExportObjectUrl: The full Store-Relative Form URL for the list item.

Ordinal: An identifier of the list item that is a number used with the @LastOrdinal parameter. All rows returned MUST have an ordinal value greater than @LastOrdinal. The ordinals of all rows returned MUST be in increasing order.

ParentId: The identifier of the list containing this list item.

DirName: The **Directory Name** of the list item.

LeafName: The **Leaf Name** of the list item.

IntId: The identifier for the list item, uniquely identifying it within the AllUserData Table, specified in [\[MS-WSSFO\]](#) section 2.2.7.3 AllUserData Table.

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 which have more application-defined columns of one or more data types than can fit in a single row in the AllUserData Table, specified in [\[MS-WSSFO\]](#) section 2.2.7.3 AllUserData Table. This Result Set MUST contain only rows with tp_RowOrdinal equal to 0.

tp_UIVersion: The user interface (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_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 date/time value in UTC specifying when this list item was last modified.

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

tp_Ordering: A concatenation of timestamp values in yyyyMMddHHmmss (year, month, day, hour, minute, second) format, specifying the threading structure of the list items in a legacy discussion board list (a list with a List Base Type, specified in [\[MS-WSSFO\]](#) section 2.2.3.9 List Base Type, of 3). For list items in all other types of list, this parameter MUST be NULL.

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

tp_ModerationStatus: A moderation status 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 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.

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

tp_CopySource: The URL used as a source for the list item. If this list item is 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. This bit is set to 0, if the list item does not have a destination location set. This is NULL, if the list item never had a destination location.

tp_AuditFlags: An **audit flags** 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_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.

tp_ThreadIndex: A binary structure specifying the list item's position within a legacy discussion board list (a list with a List Base Type, specified in [MS-WSSFO] section 2.2.3.9 List Base Type, of 3). For list items in all other types of list, this parameter MUST be NULL.

tp_UIVersionString: A calculated column containing the value of the tp_UIVersion column as a Displayed Version string.

Id: The identifier of the Document associated with this list item.

Type: An integer identifier specifying the Document Store Type, specified in [MS-WSSFO] section 2.2.2.1 Document Store Type, of the Document.

FolderId: The document identifier of the Document's parent container.

Acl: The binary serialization of the WSS ACL Format access control list, specified in [\[MS-WSSFO\]](#) section 2.2.4.6 WSS ACL Format, for this Document. This is either explicitly defined or inherited from the parent object.

UserLoginName: If @ListId is the identifier of the user information list and the list item is a user, then UserLoginName MUST be the Login Name for the **Principal**. Otherwise, this MUST be NULL.

GroupName: If @ListId is the identifier of the user information list and the list item is a group, then GroupName MUST be the name of the group. Otherwise, this MUST be NULL.

tp_Level: A **Publishing Level** value specifying the publishing status of this version of the list item.

3.1.4.30.2 Field Data Result Set

The Field Data Result Set returns Field data for the current versions of the list items in the specified list. If @VersionSetting is 1, only list item information with a Publishing Level of Published will be

returned. Otherwise, data for one instance of each unique list item will be returned, except information about versions with a Publishing Level of Checked Out.

The Field Data Result Set MUST return the first 200 list items with export ordinals greater than @LastOrdinal, or all the remaining list items with export ordinals greater than @LastOrdinal, whichever is less. The Field Data Result Set is defined using T-SQL syntax, as follows:

```
Id                uniqueidentifier,  
IntId             int,  
Id                uniqueidentifier,  
tp_UIVersion      int,  
tp_UIVersionString nvarchar(61),  
tp_RowOrdinal     int,  
<Fields specified by @Columns> <types will vary>,  
tp_Level          tinyint;
```

Id: Identifier used by the export process to refer to the list item.

IntId: The identifier for the list item, uniquely identifying it within the AllUserData Table, specified in [\[MS-WSSFO\]](#) section 2.2.7.3 AllUserData Table.

Id: The identifier of the Document associated with this list item.

tp_UIVersion: The user interface (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_UIVersionString: A calculated column containing the value of the tp_UIVersion column as a Displayed Version string.

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 which have more application-defined columns of one or more data types than can fit in a single row in the AllUserData Table, specified in [\[MS-WSSFO\]](#) section 2.2.7.3 AllUserData Table.

<Fields specified by @Columns>: This part of the Result Set MUST contain one column for each column specified in the @Columns parameter. The data returned will be from the application-defined fields of the AllUserData Table, specified in [\[MS-WSSFO\]](#) section 2.2.7.3 AllUserData Table, that matches the names specified in the @Columns parameter.

tp_Level: A **Publishing Level** value specifying the publishing status of this version of the list item.

3.1.4.30.3 Multi Lookup Result Set

The Multi Lookup Result Set returns the current version of the information about list items that lookup multiple values in another list. If @VersionSetting is 1, only list item information with a Publishing Level of Published will be returned. Otherwise, data for one instance of each unique list item will be returned, except information about versions with a Publishing Level of Checked Out.

The Multi Lookup Result Set will return information about the first 200 list items to be exported with Export Ordinals greater than @LastOrdinal, or if there are less than 200 list items with Export Ordinals greater than @LastOrdinal, information about all of them MUST be returned. The Result Set MUST be ordered by the Export Ordinal, then tp_FieldId, and then tp_Ordinal. The Multi Lookup Result Set is defined using T-SQL syntax, as follows:

```
Id                uniqueidentifier,  
Id                uniqueidentifier,
```

```

tp_FieldId      uniqueidentifier,
IntId           int,
tp_UIVersion    int,
tp_Level        tinyint,
tp_Ordinal      int;

```

Id: The identifier used by the Export Operation to refer to the list item.

Id: The document identifier of the Document associated with the list item containing the lookup.

tp_FieldId: The identifier of the Field with the multi-value lookup.

IntId: The identifier in the AllUserData Table, specified in [\[MS-WSSFO\]](#) section 2.2.7.3 AllUserData Table, for the list item containing the data being looked up.

tp_UIVersion: The user interface (UI) version number. The default value of tp_UIVersion is 512, which corresponds to a **Displayed Version** of 1.0.

tp_Level: A **Publishing Level** value specifying the publishing status of this version of the list item.

tp_Ordinal: Ordinal number used to determine the ordering of the values returned in the IntId column.

3.1.4.30.4 List Item Versions Result Set

The List item Versions Result Set returns information about the list items to be exported. If @VersionSetting is 4, information about all versions of the list items will be returned. Otherwise, only information about the current version will be returned. Information about versions of list items with a Publishing Level of Checked Out MUST NOT be returned. The List Item Versions Result Set MUST return the first 200 list items with Export Ordinals greater than @LastOrdinal, or if there are less than 200 list items with Export Ordinals greater than @LastOrdinal, information about all of the remaining list items MUST be returned. The Result Set must be ordered first by the Export Ordinal, and then tp_UIVersion. The List Item Versions Result Set is defined using T-SQL syntax, as follows:

```

Id                uniqueidentifier,
ExportObjectUrl   nvarchar(260),
Ordinal           int,
ParentId          uniqueidentifier,
DirName           nvarchar(256),
LeafName          nvarchar(128),
IntId             int,
tp_RowOrdinal     int,
tp_UIVersion      int,
tp_Author         int,
tp_Editor         int,
tp_Modified       datetime,
tp_Created        datetime,
tp_Ordering       varchar(512),
tp_HasAttachment  bit,
tp_ModerationStatus int,
tp_IsCurrent      bit,
tp_ItemOrder      float,
tp_InstanceID     int,
tp_CopySource     nvarchar(260),
tp_HasCopyDestinations bit,
tp_AuditFlags     int,

```

```

tp_InheritAuditFlags    int,
tp_ContentType          nvarchar(255),
tp_ContentTypeId       varbinary(512),
{tp_ThreadIndex}       varbinary(512),
tp_UIVersionString     nvarchar(61),
{Id}                   uniqueidentifier,
{Type}                 tinyint,
{Acl}                  image,
{UserLoginName}        nvarchar(255),
{GroupName}            nvarchar(255),
tp_Level               tinyint;

```

Id: Identifier used by the Export Operation to refer to the list item.

ExportObjectUrl: The full Store-Relative Form URL for the list item.

Ordinal: The Export Ordinal of the list item. All rows returned MUST have an ordinal value greater than @LastOrdinal.

ParentId: The identifier of the list containing this list item.

DirName: The Directory Name of the list item.

LeafName: The Leaf Name of the list item.

IntId: The identifier for the list item, uniquely identifying it within the AllUserData Table, specified in [\[MS-WSSFO\]](#) section 2.2.7.3 AllUserData Table.

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 which have more application-defined columns of one or more data types than can fit in a single row in the AllUserData Table, specified in [\[MS-WSSFO\]](#) section 2.2.7.3 AllUserData Table.

tp_UIVersion: The user interface (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_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: The time value in UTC specifying when this list item was last modified.

tp_Created: The time value in UTC specifying when this list item was created.

tp_Ordering: A concatenation of timestamp values in yyyyMMddHHmmss format (year, month, day, hour, minute, second), specifying the threading structure of the list items in a legacy discussion board list (a list with a List Base Type, specified in [\[MS-WSSFO\]](#) section 2.2.3.9 List Base Type, of 3). For list items in all other types of list, this parameter MUST be NULL.

tp_HasAttachment: This MUST be 1 if the list item has an **attachment** associated with it, otherwise this MUST be 0.

tp_ModerationStatus: A moderation status value indicating the current moderation approval status of the list item.

tp_IsCurrent: This MUST be 1 if this is a current version of this list item, otherwise this MUST be 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.

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

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

tp_HasCopyDestinations: This is set to 1 if destination locations for the list item to be copied to have been set. This is set to 0 if the list item does not have a destination location set. This is NULL if the list item never had a destination location.

tp_AuditFlags: An **audit flags** 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_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.

{tp_ThreadIndex}: This value MUST be NULL and MUST be ignored.

tp_UIVersionString: A calculated column containing the value of the tp_UIVersion column as a Displayed Version string.

{Id}: This value MUST be NULL and MUST be ignored.

{Type}: This value MUST be NULL and MUST be ignored.

{Acl}: This value MUST be NULL and MUST be ignored.

{UserLoginName}: This value MUST be NULL and MUST be ignored.

{GroupName}: This value MUST be NULL and MUST be ignored.

tp_Level: A **Publishing Level** value specifying the publishing status of this version of the list item.

3.1.4.30.5 Versions Fields Data Result Set

The Versions Fields Data Result Set returns Field data for the list items to be exported. If @VersionSetting is 4, information about all versions of the list items will be returned. Otherwise, only information about the current version will be returned. The Result Set MUST NOT contain information for versions of list items that have a Publishing Level of Checked Out.

The Versions Fields Data Result Set MUST return the first 200 list items with Export Ordinals greater than @LastOrdinal, or if there are less than 200 list items with Export Ordinals greater than @LastOrdinal, information about all of them MUST be returned. The Result Set MUST be ordered first by the Export Ordinal, and then tp_UIVersion. The Versions Fields Data Result Set is defined using T-SQL syntax, as follows:

Id	uniqueidentifier,
IntId	int,
DocId	uniqueidentifier,
tp_UIVersion	int,
tp_UIVersionString	nvarchar(61),
tp_RowOrdinal	int,

```
<Fields specified by @Columns>    <types will vary>,  
tp_Level                          tinyint;
```

Id: The identifier used by the Export Operation to refer to the list item.

IntId: The identifier for the list item uniquely identifying it within the AllUserData Table, specified in [\[MS-WSSFO\]](#) section 2.2.7.3 AllUserData Table.

DocId: The document identifier of the **Document associated with this list item.**

tp_UIVersion: The user interface (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_UIVersionString: A calculated column containing the value of the tp_UIVersion column as a Displayed Version string.

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 which have more application-defined columns of one or more data types than can fit in a single row in the AllUserData Table, specified in [\[MS-WSSFO\]](#) section 2.2.7.3 AllUserData Table.

<Fields specified by @Columns>: This part of the Result Set **MUST** contain one column for each column specified in the @Columns parameter. The data returned will be from the application-defined fields of the AllUserData Table, specified in [\[MS-WSSFO\]](#) section 2.2.7.3 AllUserData Table, that matches the names specified in the @Columns parameter.

tp_Level: A **Publishing Level** value specifying the publishing status of this version of the list item.

3.1.4.30.6 Versions Multi Lookup Result Set

The Versions Multi Lookup Result Set returns the information about list items that lookup multiple values in another list. If @VersionSetting is 4, information about all versions of the list items will be returned. Otherwise, only information about the current version will be returned. The Result Set **MUST NOT** contain information for versions of list items that have a Publishing Level of Checked Out.

The Versions Multi Lookup Result Set will return information about the first 200 list items to be exported with Export Ordinals greater than @LastOrdinal, or if there are less than 200 list items with Export Ordinals greater than @LastOrdinal, information about all of them **MUST** be returned. The Result Set **MUST** be ordered by the Export Ordinal, then tp_FieldId, and then tp_Ordinal. The Versions Multi Lookup Result Set is defined using T-SQL syntax, as follows:

```
Id                          uniqueidentifier,  
DocId                       uniqueidentifier,  
tp_FieldId                  uniqueidentifier,  
IntId                       int,  
tp_UIVersion                int,  
tp_Level                    tinyint,  
tp_Ordinal                  int;
```

Id: The identifier used by the Export Operation to refer to the list item.

DocId: The document identifier of the Document associated with the list item containing the lookup.

tp_FieldId: The identifier of the Field with the multi-value lookup.

IntId: The identifier in the AllUserData Table, specified in [\[MS-WSSFO\]](#) section 2.2.7.3 AllUserData Table, for the list item containing the data being looked up.

tp_UIVersion: The user interface (UI) version number. The default value of tp_UIVersion is 512, which corresponds to a **Displayed Version** of 1.0.

tp_Level: A **Publishing Level** value specifying the publishing status of this version of the list item.

tp_Ordinal: Ordinal number used to determine the ordering of the values returned in the IntId column.

3.1.4.30.7 List Item Links Result Set

The List Item Links Result Set returns information about the links, in server-relative form, in the list items to be exported. If @VersionSetting is 1, only list item information with a Publishing Level of Published will be returned. Otherwise, data for all versions of list item information with a Publishing Level that is not Checked Out will be returned.

The List Item Links Result Set will return information for the first 200 list items with export ordinals greater than @LastOrdinal, or all list items with export ordinals greater than @LastOrdinal, whichever is less. The List Item Links Result Set is defined using T-SQL syntax, as follows:

```
Id                uniqueidentifier,  
{TargetDocId}    uniqueidentifier,  
TargetUrl        nvarchar(261);
```

Id: Identifier used by the Export Operation to refer to the list item.

{TargetDocId}: The identifier of the Document that the link points to. If the Document is not found, a new identifier will be generated.

TargetUrl: Server-relative URL of the link. This MUST start with a slash.

3.1.4.30.8 Attachments Result Set

The Attachments Result Set returns information about the attachments in the specified list items to export. This Result Set MUST only be returned if @ListId matches an existing list identifier and the Document associated with the list's URL can be found. The Attachments Result Set will return information for the first 200 list items with Export Ordinal values greater than @LastOrdinal, or all list items with Export Ordinal values greater than @LastOrdinal, whichever is less. The Result Set MUST be ordered by Export Ordinal. The Attachments Result Set is defined using T-SQL syntax, as follows:

```
ListItemId        uniqueidentifier,  
DocId            uniqueidentifier,  
WebId            uniqueidentifier,  
ListId           uniqueidentifier,  
ScopeId         uniqueidentifier,  
MetaInfo        varbinary(max),  
TimeCreated      datetime,  
TimeLastModified datetime,  
Size            int,  
DirName         nvarchar(256),  
LeafName        nvarchar(128),  
Url            nvarchar(260);
```

ListItemId: The list item identifier of the list item that's used by the export process.

DocId: The document identifier of the Document.

WebId: The site identifier for the Site containing this Document.

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

ScopeId: The identifier **of the Scope which contains the Document.**

MetaInfo: A **Metadict** for the Document. The Metadict format is specified in [\[MS-FPSE\]](#), section 2.2.2.5.3.

TimeCreated: Date/time value in UTC specifying when this Document was created.

TimeLastModified: A datetime containing a timestamp in UTC specifying when the Document was last saved. This value does not necessarily correspond to the actual time when the Document was last modified.

Size: The number of bytes in the **Document Stream.**

DirName: The Directory Name of the Document.

LeafName: The Leaf Name of the Document.

Url: The full Store-Relative Form URL for the specified Document.

3.1.4.30.9 Missing List Result Set

The Missing List Result Set MUST only be returned if @ListId does not match an existing list identifier or if the Document associated with the list's URL cannot be found. This result set MUST have 0 rows. The Missing List Result Set is defined using T-SQL syntax, as follows:

Id	uniqueidentifier,
DocId	uniqueidentifier,
WebId	uniqueidentifier,
ListId	uniqueidentifier,
ScopeId	uniqueidentifier,
MetaInfo	varbinary(max),
TimeCreated	datetime,
TimeLastModified	datetime,
Size	int,
DirName	nvarchar(256),
LeafName	nvarchar(128),
Url	nvarchar(260);

Id: Identifier of the Document.

DocId: The document identifier of the Document.

WebId: The site identifier for the Site containing this Document.

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

ScopeId: The identifier of the Scope which contains the Document.

MetaInfo: A **Metadict** for the Document. The Metadict format is specified in [\[MS-FPSE\]](#), section 2.2.2.5.3.

TimeCreated: Date/time value in UTC specifying when this Document was created.

TimeLastModified: A datetime containing a timestamp in UTC specifying when the Document was last saved. This value does not necessarily correspond to the actual time when the Document was last modified.

Size: The number of bytes in the **Document Stream**.

DirName: The Directory Name of the Document.

LeafName: The Leaf Name of the Document.

Url: The full Store-Relative Form URL for the specified Document.

3.1.4.30.10 List Item Event Receivers Result Set

The List Item Event Receivers Result Set returns information about the event receivers defined for the list items to be exported. The List Item Event Receivers Result Set **MUST** contain information for the first 200 list items with export ordinals greater than @LastOrdinal, or all list items with export ordinals greater than @LastOrdinal, whichever is less. The List Item Event Receivers Result Set is defined using T-SQL syntax, as follows:

```
Id                uniqueidentifier,
HostId            uniqueidentifier,
WebId             uniqueidentifier,
Name              nvarchar(256),
HostType          int,
ItemId           int,
Type             int,
SequenceNumber   int,
Assembly          nvarchar(256),
Class            nvarchar(256),
Data             nvarchar(256),
Filter           nvarchar(256),
Credential        int,
ListItemId       uniqueidentifier;
```

Id: The identifier of the event receiver.

HostId: The identifier of the event host that the event receiver is associated with.

WebId: The site identifier for the Site that contains the event receiver.

Name: The name of the event receiver.

HostType: The type of the event host that the event receiver is associated with. HostType **MUST** be a value of event host type, specified in [\[MS-WSSFO\]](#) section 2.2.3.5 Event Host Type.

ItemId: Reserved. A client **MUST NOT** use ItemId.

Type: The type of the event receiver. Type **MUST** be a value of event receiver type, specified in [\[MS-WSSFO\]](#) section 2.2.3.6 event receiver Type.

SequenceNumber: The sequence number of the event receiver. SequenceNumber MUST be greater than OR equal to 0 AND less than OR equal to 65535.

Assembly: The assembly name of the implementation of the event receiver.

Class: The fully qualified class name of 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: Unused and MUST be ignored.

Credential: Unused and MUST be ignored.

ListItemId: Identifier used by the Export Operation to refer to the list item in the table of export objects.

3.1.4.31 proc_DeplGetLookupItemData

The proc_DeplGetLookupItemData stored procedure is called to return information about list items in a given list that are referenced by other lists. proc_DeplGetLookupItemData is defined using T-SQL syntax, as follows:

```
PROCEDURE proc_DeplGetLookupItemData (
    @SiteId          uniqueidentifier,
    @ListId          uniqueidentifier
);
```

@SiteId: The site collection identifier for the **Site Collection** containing the list items.

@ListId: The list identifier for the list containing the list items.

Return Code Values: An integer which the protocol client MUST ignore.

Result Sets: MUST return the following Result Set:

3.1.4.31.1 Lookup Data Result Set

The Lookup Data Result Set returns information about list items that are referenced by other lists. The Lookup Data Result Set is defined using T-SQL syntax, as follows:

```
Id          int,
Id          uniqueidentifier,
{URL}      nvarchar(261),
Included   bit;
```

Id: The identifier for the list item, uniquely identifying it within the AllUserData Table, specified in [\[MS-WSSFO\]](#) section 2.2.7.3 AllUserData Table.

Id: The identifier of the Document associated with this list item.

{URL}: The Server-relative URL of this Document. This MUST begin with a slash.

Included: If this Document is in the table of export objects, this MUST be 1. Otherwise, this MUST be 0.

3.1.4.32 proc_DeplGetLookupListUrl

The proc_DeplGetLookupListUrl stored procedure is called to retrieve the URL of the list. proc_DeplGetLookupListUrl is defined using T-SQL syntax, as follows:

```
PROCEDURE proc_DeplGetLookupListUrl(  
    @SiteId          uniqueidentifier,  
    @ListId          uniqueidentifier,  
    @Url             nvarchar(260) OUTPUT  
);
```

@SiteId: The site collection identifier for the **Site Collection** containing the list.

@ListId: The list identifier for the list.

@Url: This will output the Server-relative URL of the list. The output **MUST** start with a slash, unless the list is not found. If the list is not found, this output value **MUST** be the same as its input value.

Return Code Values: An integer which the protocol client **MUST** ignore.

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

3.1.4.33 proc_DeplGetNextObjectBatch

The proc_DeplGetNextObjectBatch stored procedure is called during an Export Operation to select the next set of Export Objects that need to be exported. proc_DeplGetNextObjectBatch is defined using T-SQL syntax, as follows:

```
PROCEDURE proc_DeplGetNextObjectBatch(  
    @LastOrdinal     int  
);
```

@LastOrdinal: All Export Objects returned by proc_DeplGetNextBatch **MUST** have values in the Ordinal column greater than @LastOrdinal. When calling proc_DeplGetNextObjectBatch for the first time, set @LastOrdinal to 0. When calling proc_DeplGetNextObjectBatch to retrieve more Objects set @LastOrdinal to the largest Ordinal value returned in the previous call to proc_DeplGetNextObjectBatch.

Return Code Values: An integer which the protocol client **MUST** ignore.

Result Sets: **MUST** return the following result set:

3.1.4.33.1 Export Object Batch Result Set

Export Object Batch Result Set returns the next set of export objects to be exported. The Export Object Batch Result Set **MUST** return 100 rows or one row for each object to be exported whichever is smaller. The Export Object Batch Result Set is defined using T-SQL syntax, as follows:

```
Id                uniqueidentifier,  
Type              tinyint,  
ParentId          uniqueidentifier,  
ParentWebId       uniqueidentifier,  
{FullUrl}         nvarchar(260)  
Ordinal           int,  
IsDeleted         tinyint,
```

ContentTypeId	varbinary (512),
{ParentWebUrl}	nvarchar (260),
EventTime	datetime,
ChangeLogId	bigint;

Id: The identifier of the object to be exported.

Type: A value that specifying the [deployment object type](#), specified in section 2.2.3.1, of the object stored in the row.

ParentId: The identifier of the parent of the object.

ParentWebId: The site identifier for the Site that the object lives in. For an Object of type root Site this MUST be NULL.

{FullUrl}: Server-relative URL of the Export Object.

Ordinal: The identifier of the Object used with the @LastOrdinal parameter. All rows returned MUST have an ordinal value greater than @LastOrdinal. The ordinals of all rows returned MUST be in increasing order.

IsDeleted: Whether the object was deleted. This is set to 1 if the Object was deleted; otherwise set to 0. Used only during an incremental export operation otherwise this MUST be ignored.

ContentTypeId: A Content Type identifier value describing the valid folder content types that this Export Object can be used to view. If this Export Object is not restricted to a particular content type, this value MAY be 0x or NULL.

{ParentWebUrl}: Server-relative URL of the Site that the object lives in. For an object of type root site, this MUST be NULL.

EventTime: The UTC time from the change log that the object was modified. Used only during an incremental export operation, otherwise this MUST be ignored.

ChangeLogId: The identifier of the Event. Used only during an incremental export operation, otherwise this MUST be ignored.

3.1.4.34 proc_DeplGetParentWebIdForList

The proc_DeplGetParentWebIdForList stored procedure is called to return information about the Site that contains the given list. proc_DeplGetParentWebIdForList is defined using T-SQL syntax, as follows:

```
PROCEDURE proc_DeplGetParentWebIdForList (
    @ListId          uniqueidentifier
);
```

@ListId: The list identifier for the list.

Return Code Values: An integer which the protocol client MUST ignore.

Result Sets: If the input parameter is NULL then it MUST NOT return any Result Set. Otherwise, if the parameter is NOT NULL, then it MUST return the following Result Set:

3.1.4.34.1 Parent Site Result Set

The Parent Site Result Set returns information about the Site containing the list. The Parent Site Result Set **MUST** contain exactly one row if the list exists. Otherwise, it **MUST** contain 0 rows. The Parent Site Result Set is defined using T-SQL syntax, as follows:

```
tp_WebId          uniqueidentifier;
```

tp_WebId: The site identifier for the Site containing the list.

3.1.4.35 proc_DeplGetParentWebIdFromDocId

The proc_DeplGetParentWebIdFromDocId stored procedure is called to return information about the Site and list that contain a given Document. proc_DeplGetParentWebIdFromDocId is defined using T-SQL syntax, as follows:

```
PROCEDURE proc_DeplGetParentWebIdFromDocId(  
    @DocId          uniqueidentifier  
);
```

@DocId: The document identifier of the Document.

Return Code Values: An integer which the protocol client **MUST** ignore.

Result Sets: If at least one of the input parameters is NULL then it **MUST NOT** return any Result Set. Otherwise, if both parameters are NOT NULL, then it **MUST** return the following Result Set:

3.1.4.35.1 Document Container Result Set

The Document Container Result Set returns information about the Site and list that contain the Document. The Document Container Result Set **MUST** contain one row if the Document exists. Otherwise, it **MUST** contain 0 rows. The Document Container Result Set is defined using T-SQL syntax, as follows:

```
WebId          uniqueidentifier,  
ListId         uniqueidentifier;
```

WebId: The site identifier for the **Site** containing the Document.

ListId: The list identifier of the list containing the Document. If the Document is not contained in a list, this **MUST** be NULL.

3.1.4.36 proc_DeplGetParentWebIdFromDocUrl

The proc_DeplGetParentWebIdFromDocUrl Stored Procedure is called to return the location of a specified **Document**. The Stored Procedure is defined using T-SQL syntax as follows:

```
PROCEDURE proc_DeplGetParentWebIdFromDocUrl (  
    @SiteId          uniqueidentifier,  
    @DocUrl          nvarchar(260)  
);
```

@SiteId: The site collection identifier for the **Site Collection** that contains a Document with the @DocUrl parameter value.

@DocUrl: A **Store-Relative** Form URL for a **Document** within the Site.

Return Code Values: An integer which the protocol client MUST ignore.

Result Sets: If at least one of the input parameters is NULL then it MUST NOT return any Result Set. Otherwise, if both parameters are NOT NULL, then it MUST return the following Result Set:

3.1.4.36.1 Location Result Set

The Location Result Set contains information about the location of the document within the site collection. The Location Result Set MUST contain one row if the document is found within the site collection, and MUST contain 0 rows otherwise. The Location Result Set is defined using T-SQL syntax, as follows:

```
WebId          uniqueidentifier,  
ListId         uniqueidentifier;
```

WebId: The site identifier for the **Site**.

ListId: The list identifier of the list in the Site which contains the specified Document. If the Document is not in any list, this MUST be NULL.

3.1.4.37 proc_DeplGetRootObjects

The proc_DeplGetRootObjects stored procedure is called to return information about the root level objects (objects that do not have their parent objects included as part of the export) to be processed during an Export Operation. proc_DeplGetRootObjects is defined using T-SQL syntax, as follows:

```
PROCEDURE proc_GetRootObjects ();
```

Return Code Values: An integer which the protocol client MUST ignore.

Result Sets: MUST return the following Result Set:

3.1.4.37.1 Root Objects Result Set

The Root Objects Result Set returns information about objects being exported that do not have their parent objects also included as part of the export. The Root Objects Result Set is defined using T-SQL syntax, as follows:

```
Id              uniqueidentifier,  
Type            tinyint,  
ParentId       uniqueidentifier,  
FullUrl        nvarchar(256),  
FullUrl        nvarchar(260),  
IsDependency   tinyint,  
IsSelected     tinyint;
```

Id: Identifier used by the Export Operation to refer to the object being exported.

Type: An identifier specifying the object's [deployment object type](#), specified in section 2.2.3.1.

ParentId: Identifier of the parent of this object.

FullUrl: The Store-Relative Form URL of the Site containing this object.

FullUrl: The Store-Relative Form URL of this object.

IsDependency: This MUST be 1 if and only if the object is referenced from another object being exported using a Link. Otherwise, this MUST be 0.

IsSelected: This MUST be 1 if this object was chosen directly by the user for export. Otherwise, this MUST be 0.

3.1.4.38 **proc_DeplGetSystemObjects**

The `proc_DeplGetSystemObjects` stored procedure is called to return information about Migration System Objects that are referenced by objects to be exported. `proc_DeplGetSystemObjects` is defined using T-SQL syntax, as follows:

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

@SiteId: The site collection identifier for the **Site Collection** containing the objects. MUST be the identifier of an existing Site Collection.

Return Code Values: An integer which the protocol client MUST ignore.

Result Sets: MUST return the following Result Set:

3.1.4.38.1 **System Objects Result Set**

The System Objects Result Set returns information about Migration System Objects that are referenced by objects to be exported. The System Objects Result Set MUST NOT contain identical rows.

The System Objects Result Set is defined using T-SQL syntax, as follows:

```
Id          uniqueidentifier,  
Type        tinyint,  
{Url}      nvarchar(261);
```

Id: The identifier of the object.

Type: An identifier specifying the object's [deployment object type](#), specified in section 2.2.3.1.

{Url}: The Server-relative URL of the object. This MUST start with a slash.

3.1.4.39 **proc_DeplOrderExportObjects**

The `proc_DeplOrderExportObjects` stored procedure is called to sort the table of export objects. `proc_DeplOrderExportObjects` is defined using T-SQL syntax, as follows:

```
PROCEDURE proc_DeplOrderExportObjects(  

```

```

        @TestRun            tinyint,
        @ObjectCount        int            OUTPUT
    );

```

@TestRun: If this is 1, proc_DeplOrderExportObjects MUST only output a value for @ObjectCount and do nothing else. If @TestRun is not 1, objects to be exported are ordered by [deployment object type](#) specified in section 2.2.3.1, URL, and the time they were updated.

@ObjectCount: Outputs the number of entries in the table of export objects.

Return Code Values: An integer which the protocol client MUST ignore.

Result Sets: MUST NOT return any result sets.

3.1.4.40 proc_DeplReserveNextAvailableId

The proc_DeplReserveNextAvailableId stored procedure is called to reserve a list item identifier in a given list during an Import Operation. proc_DeplReserveNextAvailableId is defined using T-SQL syntax, as follows:

```

PROCEDURE proc_DeplReserveNextAvailableId(
    @WebId            uniqueidentifier,
    @ListId           uniqueidentifier,
    @ListItemId       int
);

```

@WebId: The site identifier for the Site containing the list.

@ListId: The list identifier for the list.

@ListItemId: The list item identifier of the list item to reserve. There MUST be no action taken if @ListItemId is already in use.

Return Code Values: An integer which the protocol client MUST ignore.

Result Sets: MUST NOT return any result sets.

3.1.4.41 proc_DeplUpdateWebProperties

The proc_DeplUpdateWebProperties Stored Procedure is called to update various stored values of a given Site during an Import Operation. proc_DeplUpdateWebProperties is defined using T-SQL syntax, as follows:

```

PROCEDURE proc_DeplUpdateWebProperties(
    @SiteId            uniqueidentifier,
    @WebId             uniqueidentifier,
    @AlternateCSSUrl   nvarchar(260),
    @CustomizedCssFiles nvarchar(260),
    @CustomJSUrl       nvarchar(260),
    @ExternalSecurityProvider uniqueidentifier,
    @ThemeComposite    nvarchar(64),
    @TimeCreated        datetime,
    @WebTemplateId     int,
    @ProvisionConfig    int
);

```


);

@SiteId: This value MUST be ignored.

@WebId: The site identifier for the Site. If the Site is not found, `proc_DeplUpdateWebProperties` MUST NOT update anything.

@AlternateCSSUrl: New value for the URL for a custom cascading style sheet file registered on the Site for use in Pages of the Site. If this is "", the stored value MUST be set to NULL. If this is NULL, the stored value MUST NOT be changed.

@CustomizedCssFiles: New value of an implementation-specific list of custom cascading style sheet files associated with this Site. If this is "", the stored value MUST be set to NULL. If this is NULL, the stored value MUST NOT be changed.

@CustomJSUrl: New value of the URL for a custom JavaScript file registered on the Site for use in Pages of the Site. If this is "", the stored value MUST be set to NULL. If this is NULL, the stored value MUST NOT be changed.

@ExternalSecurityProvider: New value of the COM class of the external **security provider (1)** for this site. If this is '00000000-0000-0000-0000-000000000000', the stored value MUST be set to NULL and the Site MUST use the native security implementation. If this is NULL, the stored value MUST NOT be changed.

@ThemeComposite: New value of the name of a **theme** that is used as part of the display of the Site. If this is "", the stored value MUST be set to NULL. If this is NULL, the stored value MUST NOT be changed.

@TimeCreated: New value of the timestamp in UTC specifying when this Site was created. If this is NULL, the timestamp MUST NOT be changed.

@WebTemplateId: New value of the identifier of the **site definition** that contains the **site template** configuration used to provision this Site. If this is NULL, the stored value MUST NOT be changed.

@ProvisionConfig: New value of the identifier of the Site Template Configuration used to provision this Site. If this is NULL, the stored value MUST NOT be changed.

Return Code Values: An integer which the protocol client MUST ignore.

Result Sets: MUST NOT return any result sets.

3.1.4.42 `proc_FileExists`

The `proc_FileExists` **Stored Procedure** is called to search if a **Document** exists in a **Site Collection**. `proc_FileExists` is defined using T-SQL syntax, as follows:

```
PROCEDURE proc_FileExists (
    @SiteId                uniqueidentifier,
    @WebId                 uniqueidentifier,
    @DocId                 uniqueidentifier,
    @Url                   nvarchar(260),
    @DeleteRecycleBinItems bit= 0,
    @IsDeleted             bit OUTPUT,
    @DocExists             bit OUTPUT,
    @DestDocId             uniqueidentifier OUTPUT
```

);

@SiteId: The site collection identifier for the **Site Collection** which contains the specified Document.

@WebId: The site identifier for the Site used to find the specified **Document**. This parameter is used only if the @Url parameter is used to search for the **Document**.

@DocId: The document identifier of the document to find. If this parameter is NULL, the search is done using @Url parameter.

@Url: The URL of the **Document** to find. This parameter is used only if the **Document** defined by @DocId is not found in any **Site Collection** in the database, or the **Document** is found and then **removed** from the **Recycle Bin**.

@DeleteRecycleBinItems: This parameter is ignored if @DocId is NULL. If this parameter is equal to 1, and the Document exists in the Site Collection defined by @SiteId, and the Document is placed in the **Recycle Bin**, then the Document defined by @DocId will be **removed** from the **Recycle Bin**.

@IsDeleted: When the **Document** exists in the **Site** Collection defined by @SiteId, and the **Document** is placed in the **Recycle Bin**, the value of 1 is returned.

@DocExists: When the **Document** exists in any **Site Collection** in the database, the value of 1 is returned.

@DestDocId: If @DocId is not NULL then @DestDocId is equals to @DocId. If a **Document** is found using the @Url parameter for search, then @DestDocId is equal to the document identifier of the **Document** found.

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

Value	Description
0	Execution finished.
80	Returned when the Document specified by @DocId exists in a different Site Collection than the one defined by @SiteId, and it is found in the same database.

Result Sets: MUST NOT return any **Result Set**.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

3.2 Client Details

The **front-end Web server** acts as a client when it calls the back-end database server requesting execution of Stored Procedures.

3.2.1 Abstract Data Model

Same as section 3.1.1, [Abstract Data Model](#).

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Message Processing Events and Sequencing Rules

The **front-end Web server** handles each Stored Procedure with the same processing method of calling the Stored Procedure and waiting for the **return code** and any result sets that will be returned.

The **front-end Web server** can execute dynamically generated SQL queries against the Stored Procedures, or the Tables and Views used within the database. However, unless otherwise specified, any data addition, removal, or modification **MUST** occur only by calling the listed Stored Procedures. SQL queries **MUST NOT** attempt to add, remove, or update data in any Table or View in the Content or Configuration databases, unless explicitly described in this section.

This protocol defines the following sequences which **MUST** be followed during export operations. The protocol client needs to create the table of export objects during an Export Operation. If the Export Operation is an Incremental Export Operation, the client needs to further create an Incremental Search Scope Table. The schemas of these tables and how they are created are implementation specific.

During an Import Operation, the protocol client needs to create a [#WebParts Table](#), section 2.2.6.3. The table will be deleted when the import operation is complete.

3.2.4.1 Export Sequence

Export sequence defines the order in which a number of stored procedures **MUST** be invoked to perform an export operation. Export sequence performs the following actions.

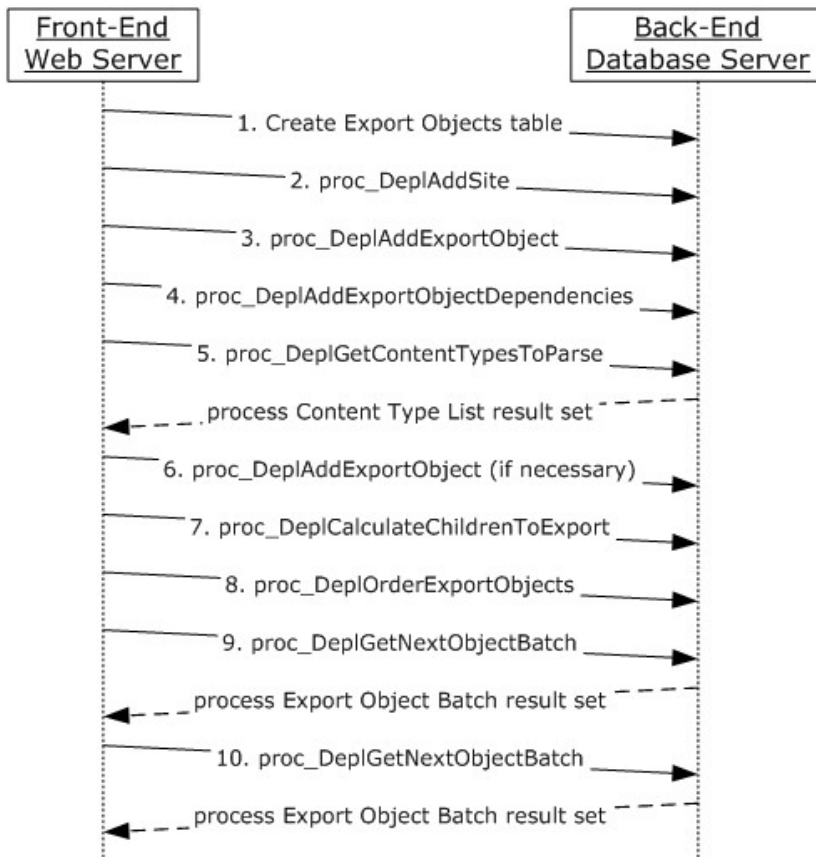


Figure 2: Create the table of export objects

1. Create the table of export objects.
2. Specify the site collection that contains the data to be exported.
3. Add objects selected by the application to the table of export objects.
4. Calculate from the objects in the table of export objects what other data they need to function properly. Add those additional objects to the table of export objects as well.
5. Retrieve the list of content types to be exported. Calculate if additional objects need to be added to the table of export objects.
6. Add additional dependency objects found in (5).
7. Calculate Add child objects (for example subsites) if application has specified that those objects need to be included.
8. Order the objects in table of export objects based on type, time, and Url.
9. Retrieve the objects from the table of export objects in a batched fashion. Serialize them into XML.
10. Repeat step 9 until there is no more objects from the table of export objects. Steps 1 through 8 MUST be carried out in a batch. Failure to do so will cause the table of export objects to be

insufficiently populated. As a result, the objects that are exported might not function properly together due to some missing objects.

3.2.4.2 Incremental Export Sequence

Incremental Export Sequence defines the order in which a number of stored procedures MUST be invoked to perform an Incremental Export Operation. Incremental Export Sequence performs the following sequences.

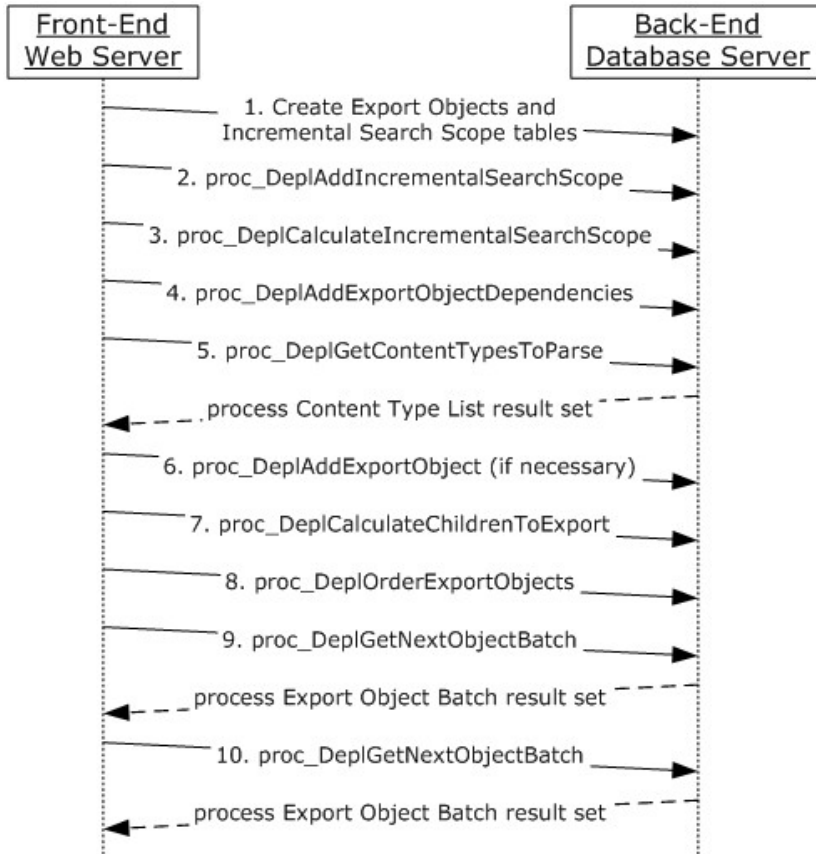


Figure 3: Create the table of export objects and the Incremental Search Scope Table

1. Create the table of export objects and the Incremental Search Scope Table.
2. Specify the sites that are included in the Incremental Export Operation.
3. Calculate any Subsites that need to be included as well.
4. Step 4 to 10 are the same as the ones illustrated in section 3.2.4.1, [Export Sequence](#).

3.2.4.3 Link Documents Export Sequence

The Link Documents Export Sequence defines the order in which a number of stored procedures MUST be called to include link documents into the table of export objects. This sequence MUST be used as part of either the [Export Sequence](#), section 3.2.4.1, or the [Incremental Export Sequence](#), section 3.2.4.2. When used in either sequence, it MUST be inserted between steps 7 and 8.

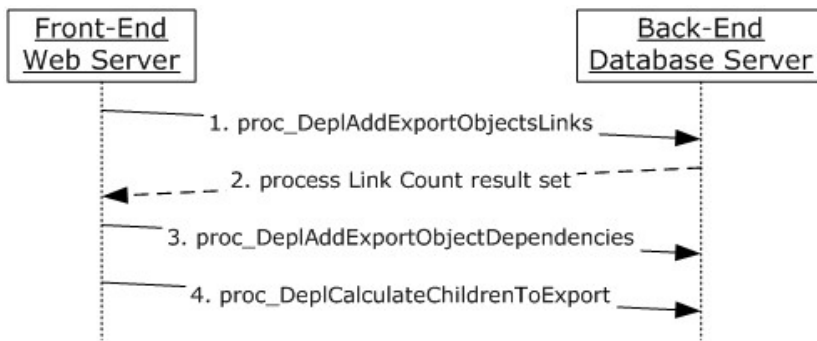


Figure 4: Link Documents Export Sequence

1. Add link documents to the table of export objects from objects that are currently in that table
2. Process the [Link Count result set](#), specified in section 3.1.4.3.1. If the Link Count result set returns one row, and {LinkCount} = 0, stop this sequence. Otherwise, continue.
3. Add additional dependency objects since new objects have been added by step 1.
4. Calculate child objects to be added to the table of export objects if the application has specified that child objects need to be included.
5. Repeat Steps 1 through 4.

3.2.4.4 Unused Web Parts Removal Sequence

To avoid dealing with duplicate web parts, the web parts for each file that is imported are stored in a temporary table so that the duplicate web parts can be removed. These web parts need to be removed because they can cause multiple web parts to be displayed.



Figure 5: Process used to remove unused Web parts

1. Create the [#WebParts Table](#), specified in section 2.2.6.3.
2. While importing the file, add the web parts referenced by this file to the #WebParts Table.
3. If the identifier of the web part is not contained in the #WebParts Table, delete any rows representing that web part in the [WebParts Table](#) and [WebPartList Table](#), sections 2.2.6.1 and 2.2.6.2.
4. Clean up all the rows in the #WebParts Table.

5. Repeat step 2, 3 and 4 for every file that is imported.

3.2.5 Timer Events

None.

3.2.6 Other Local Events

None.

4 Protocol Examples

4.1 Export a List

This example describes communication between the protocol client ("client") and the server when exporting a list, including all its content from a Site. The list URL is <http://server/subweb/MyList>, which contains one item in it. The export operation consists of the following steps.

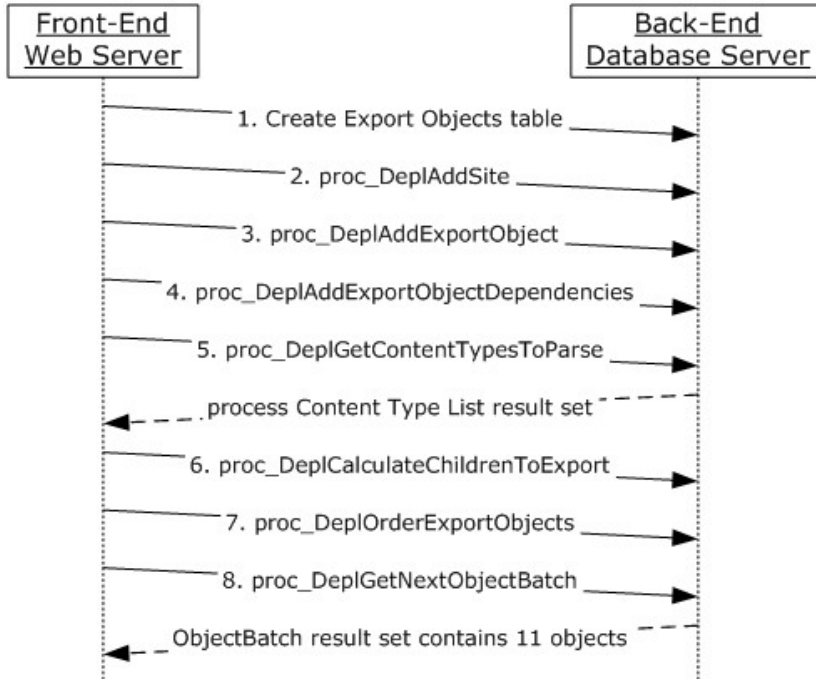


Figure 6: Process used to export a list (Part 1)

1. Create the table of export objects.
2. Specify the site collection that contains the list to be exported. [proc_DeplAddSite](#), specified in section 3.1.4.5, is called with @SiteId = '54F720B1-EC52-44CB-B347-6747DA861CBC'.
3. Add the list object to the table of export objects by calling [proc_DeplAddExportObject](#), specified in section 3.1.4.1, with the following parameters:

Parameter Name	Parameter Value
@SiteId	54F720B1-EC52-44CB-B347-6747DA861CBC
@Id	D825D4D3-1352-49BB-A498-8DC725D7458F

Parameter Name	Parameter Value
@Url	Lists/MyList
@Type	3
@IncludeDescendants	2
@IsMajorFileOnly	0
@IsLink	0
@IsSelected	1

- Add objects to the table of export objects which are needed in order for the "MyList" list to function properly. Call [proc_DeplAddExportObjectDependencies](#), specified in section 3.1.4.2, with the following parameters:

Parameter Name	Parameter Value
@SiteId	54F720B1-EC52-44CB-B347-6747DA861CBC
@IsMajorFileOnly	0
@IsLink	0

- Retrieve the list of Content Types to be exported. Call [proc_DeplGetContentTypesToParse](#), section 3.1.4.19, with the following parameter:

Parameter Name	Parameter Value
@SiteId	54F720B1-EC52-44CB-B347-6747DA861CBC

Parameter Name	Parameter Value
@LinksOnly	0

In this case, the result set does not contain any rows. Therefore, no additional objects need to be added to the table of export objects.

Add child objects to the "MyList" list, because in step 3 the @IncludeDescendants parameter specified was 2.

Parameter Name	Parameter Value
@SiteId	54F720B1-EC52-44CB-B347-6747DA861CBC
@IsMajorFileOnly	0
@IsLink	0
@IncludeUserInfo	0

1. Order the objects in table of export objects to facilitate the client to process these objects in the most efficient way. The exact sorting order is implementation specific.
2. Call [proc_DeplGetNextObjectBatch](#), specified in section 3.1.4.32, with @LastOrdinal = 0, to get the first batch of objects to export. The result set returned contains 11 rows, as shown in the following table. Data in the EventTime, ChangeLogId, and ContentTypeId columns are NULL. Data in the IsDeleted column are 0. These columns are omitted from the following table.

Id	Type	ParentId	ParentWeb Id	{FullUrl}	Ordinal	{ParentWebUrl}
54F720B1-EC52-44CB-B347-6747DA861CBC	0	NULL	NULL	NULL	1	NULL
F78F95EA-	2	36583090-	30D0C9FD-	/subweb/Lists/MyList	2	/subweb

Id	Type	ParentId	ParentWeb Id	{FullUrl}	Ordinal	{ParentWebUrl}
59FB-43A6-860A-8F2F34C06F74		D164-418C-A626-689B83092AF1	659E-4882-9C46-6CB46ABB417B			
09FE3453-98C9-4A91-B652-F9CB6E841B0B	2	F78F95EA-59FB-43A6-860A-8F2F34C06F74	30D0C9FD-659E-4882-9C46-6CB46ABB417B	/subweb/Lists/MyList/Announcement	3	/subweb
5E6B556B-68F1-488F-9B7E-839A3ED7E5F2	2	F78F95EA-59FB-43A6-860A-8F2F34C06F74	30D0C9FD-659E-4882-9C46-6CB46ABB417B	/subweb/Lists/MyList/Attachments	4	/subweb
4905FA83-DEB9-4749-BE22-5E872CCF0521	5	F78F95EA-59FB-43A6-860A-8F2F34C06F74	30D0C9FD-659E-4882-9C46-6CB46ABB417B	/subweb/Lists/MyList/AllItems.aspx	5	/subweb
457C7B49-7817-4232-B5AB-85CEB95054C5	5	F78F95EA-59FB-43A6-860A-8F2F34C06F74	30D0C9FD-659E-4882-9C46-6CB46ABB417B	/subweb/Lists/MyList/DispForm.aspx	6	/subweb
89297112-F7FF-4E36-A472-424640A77CFC	5	F78F95EA-59FB-43A6-860A-8F2F34C06F74	30D0C9FD-659E-4882-9C46-6CB46ABB417B	/subweb/Lists/MyList/EditForm.aspx	7	/subweb
0AEB3CB-97CF-4EA0-98E1-D73DBC02F66A	5	F78F95EA-59FB-43A6-860A-8F2F34C06F74	30D0C9FD-659E-4882-9C46-6CB46ABB417B	/subweb/Lists/MyList/NewForm.aspx	8	/subweb
D825D4D3-1352-49BB-A498-8DC725D7458F	3	30D0C9FD-659E-4882-9C46-6CB46ABB417B	30D0C9FD-659E-4882-9C46-6CB46ABB417B	/subweb/Lists/MyList	9	/subweb
D2DB5C42-1FB1-4B47-BE2D-85FC7B0D33AC	7	F78F95EA-59FB-43A6-860A-8F2F34C06F74	30D0C9FD-659E-4882-9C46-6CB46ABB417B	/subweb/Lists/MyList/1_.000	10	/subweb
6F7A8954-	4	D825D4D3-	30D0C9FD-	/subweb/Lists/MyList/1_.00	11	/subweb

Id	Type	ParentId	ParentWeb Id	{FullUrl}	Ordinal	{ParentWebUrl}
EC20-4861-864A-A634BCA5028D		1352-49BB-A498-8DC725D7458F	659E-4882-9C46-6CB46ABB417B	0		

What's shown in the result set is an object representing a site collection, followed by three folder objects, four file objects, one list object, one document library file object and one list item.

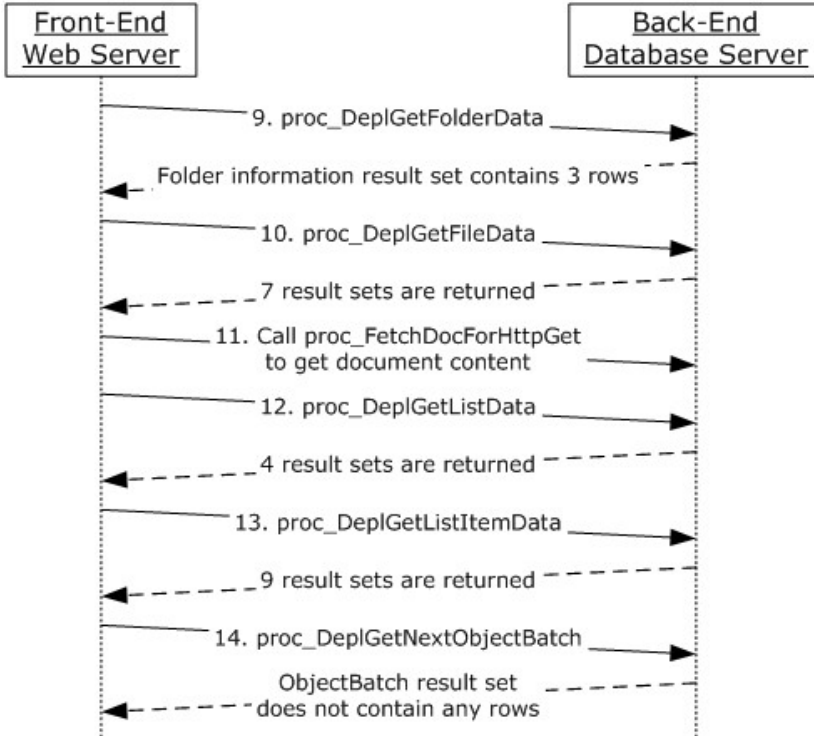


Figure 7: Process used to export a list (Part 2)

- Get additional metadata of the 3 Folder objects by calling [proc_DeplGetFolderData](#), specified in section 3.1.4.23, with the following parameters:

Parameter Name	Parameter Value
@SiteId	54F720B1-EC52-44CB-B347-6747DA861CBC
@ LastOrdinal	0

Parameter Name	Parameter Value
@VersionSetting	4

The result set contains three rows. The first one is listed in the following table:

Column Name	Description
Id	F78F95EA-59FB-43A6-860A-8F2F34C06F74
ExportObjectUrl	subweb/Lists/MyList
Ordinal	2
ParentWebId	30D0C9FD-659E-4882-9C46-6CB46ABB417B
LeafName	MyList
ListId	D825D4D3-1352-49BB-A498-8DC725D7458F
DocLibRowId	NULL
WelcomePageUrl	NULL
WelcomePageParameters	NULL
IsWebRootFolder	0

Column Name	Description
Acl	0xF3FE00001000000000000000300000003000000FFFFFFFFFFFFFF7F040000061104308B000000005000000EF12433CB0010000
MetaInfo	NULL
TimeCreated	2008-03-03 18:44:43.000
TimeLastModified	2008-03-03 18:49:33.000

- Get metadata of the next batch file objects by calling [proc_DeplGetFileData](#), specified in section 3.1.4.21, with the following parameters:

Parameter Name	Parameter Value
@SiteId	54F720B1-EC52-44CB-B347-6747DA861CBC
@ LastOrdinal	0
@VersionSetting	4

proc_DeplGetFileData returns 7 result sets:

[File Information result set](#), specified in section 3.1.4.21.1, which returns one row per file containing meta information of the file. In this sample, five rows will be returned because there are four file objects and 1 document library File object. The first row is listed in the following table:

Column Name	Description
Id	4905FA83-DEB9-4749-BE22-5E872CCF0521
ExportObjectUrl	subweb/Lists/MyList/AllItems.aspx

Column Name	Description
Ordinal	5
WebId	30D0C9FD-659E-4882-9C46-6CB46ABB417B
ListId	D825D4D3-1352-49BB-A498-8DC725D7458F
ParentId	F78F95EA-59FB-43A6-860A-8F2F34C06F74
ScopeId	6BE9822E-5AC6-41DD-940D-BF058FFA5894
DocLibRowId	NULL
MetaInfo	NULL
UIVersion	512
UIVersionString	1.0
IsGhosted	1
TimeCreated	2008-03-03 18:44:43.000
TimeLastModified	2008-03-03 18:44:43.000
DirName	subweb/Lists/MyList
LeafName	AllItems.aspx
SetupPath	pages\viewpage.aspx
SetupPathUser	REDMOND\user
SetupPathVersion	3
UnVersionedMetaInfo	NULL
CheckinComment	NULL
AuditFlags	NULL
InheritAuditFlags	NULL
Level	1
HasStream	0
Size	2661
Acl	0xF3FE000001000000000000003000000030000000FFFFFFFFFFFFFFFF7F0400000061104308B000000005000000EF12433CB0010000

- [Latest File Versions result set](#), specified in section 3.1.4.21.2, which contains five rows representing the last major version for each of the five files. All these files contain one and only one version in this example. The first row in the result set is the following.

Column Name	Description
Id	4905FA83-DEB9-4749-BE22-5E872CCF0521
UIVersion	512
TimeCreated	2008-03-03 18:44:43.000
TimeLastModified	2008-03-03 18:44:43.000
DocFlags	268
MetaInfo	NULL
CheckinComment	NULL
Level	1
SetupPath	pages\viewpage.aspx
SetupPathUser	REDMOND\user
SetupPathVersion	3
HasStream	0
ListId	D825D4D3-1352-49BB-A498-8DC725D7458F
Size	2661

- [Previous File Versions result set](#), specified in section 3.1.4.21.3, which will return one row per old version of the files given that the @VersionSettings parameter is 4. In this example, the four files have no old versions. Therefore this result set is empty.
- [File Web Part result set](#), specified in section 3.1.4.21.4, which has one row for each Web Part reference in the files returned. In this example, there are five files in total. Four of the five files contain exactly one Web Part. As a result, four rows are returned in this result set. The first one is shown as following.

Column Name	Description
tp_Id	FD655044-665A-4468-BF7D-5CD1B41E1A3A
tp_ListID	D825D4D3-1352-49BB-A498-8DC725D7458F
tp_Type	0
tp_Flags	25165825
tp_Hidden	0
tp_Threaded	0
tp_FPModified	0

Column Name	Description
tp_ReadOnly	0
tp_Scope	[Empty String]
tp_RecurrenceRowset	0
tp_ModerationType	[Empty String]
tp_Personal	0
tp_Ordered	0
tp_DisplayName	All Items
tp_ContentTypeId	0x
tp_PageUrl	/subweb/Lists/MyList/AllItems.aspx
tp_BaseViewId	1
tp_View	NULL
tp_Level	1
tp_IsDefault	0
tp_ZoneID	Main
tp_IsIncluded	1
tp_PartOrder	1
tp_FrameState	0
tp_Version	1
tp_WebPartTypeId	FD9424E7-E654-37B5-DA14-A3AD6E18C6AE
tp_AllUsersProperties	NULL
tp_PerUserProperties	NULL
tp_UserID	NULL
tp_Source	NULL
tp_CreationTime	1900-01-01 00:00:00.000
DocId	4905FA83-DEB9-4749-BE22-5E872CCF0521
tp_ListRootFolderUrl	/subweb/Lists/MyList

- [Web Part Personalization result set](#), specified in section 3.1.4.21.5. The web parts referenced in these files do not have any personalization in this example. This result set does not contain any rows.

- [File Links result set](#), specified in section 3.1.4.21.6. These files do not contain any hyperlinks to other documents within the same site collection. This result set does not contain any rows.
- [File Event Receivers result set](#), specified in section 3.1.4.21.7. These files do not have any event receiver associated with them. This result set does not contain any rows.
- Call `proc_FetchDocForHttpGet`, specified in [\[MS-WSSFO\]](#) section 3.1.4.13, to get the content of the files. In the previous step, only the file metadata was collected, but not the file stream. To get the file with export ordinal number 5, this sproc is called with the following parameters.

Parameter Name	Parameter Value
@DocSiteId	54F720B1-EC52-44CB-B347-6747DA861CBC
@DocDirName	subweb/Lists/MyList
@DocLeafName	AllItems.aspx
@LooksLikeAttachmentFile	0
@IfModifiedSince	Dec 30 1899 12:00:00:000AM
@FetchType	0
@ValidationType	0
@ClientVersion	NULL
@ClientId	NULL
@PageView	NULL

Parameter Name	Parameter Value
@FetchBuildDependencySet	0
@SystemID	0x010500000000000515000000A065CF7E784B9B5FE77C87704AB10000
@CurrentVirusVendorID	NULL
@PrefetchListScope	1
@ChunkSize	5242880
@DGCacheVersion	0
@MaxCheckinLevel	1
@HonorLevel	1
@CurrentFolderUrl	NULL

Repeat this three more times with slightly different parameters to get the streams of the other three files.

Call [proc_DeplGetListData](#), specified in section 3.1.4.25, to get the meta info of the list object. The stored procedure is called with the following parameters.

Parameter Name	Parameter Value

Parameter Name	Parameter Value
@SiteId	54F720B1-EC52-44CB-B347-6747DA861CBC
@LastOrdinal	0
@IsIncremental	0

This stored procedure returns four result sets:

[List Information result set](#), specified in section 3.1.4.25.1, which contains only one row because there is only 1 list to be exported. The row contains the following:

Column Name	Description
Id	D825D4D3-1352-49BB-A498-8DC725D7458F
ExportObjectUrl	subweb/Lists/MyList
Ordinal	9
tp_Title	MyList
tp_Description	
tp_Modified	2008-03-03 18:49:33.000
tp_Created	2008-03-03 18:44:43.000
tp_Version	0
tp_BaseType	0
tp_FeatureId	00BFEA71-D1CE-42DE-9C63-A44004CE0104
tp_ServerTemplate	104
tp_RootFolder	F78F95EA-59FB-43A6-860A-8F2F34C06F74
tp_Template	NULL
tp_ReadSecurity	1
tp_WriteSecurity	1
tp_Fields	12.0.0.6304.0.0<FieldRef Name="ContentTypeId"/><FieldRef Name="Title" ColName="nvarchar1"/><FieldRef Name="_ModerationComments" ColName="ntext1"/><FieldRef Name="File_x0020_Type" ColName="nvarchar2"/><FieldRef Name="Body" ColName="ntext2"/><FieldRef

Column Name	Description
	Name="Expires" ColName="datetime1"/> Reference
tp_Direction	0
tp_Flags	536875008
tp_ThumbnailSize	NULL
tp_WebImageWidth	NULL
tp_WebImageHeight	NULL
tp_ImageUrl	/_layouts/images/itann.gif
tp_Author	1
tp_EventSinkAssembly	NULL
tp_EventSinkClass	NULL
tp_EventSinkData	NULL
tp_EmailInsertsFolder	NULL
tp_EmailAlias	NULL
tp_DocTemplateUrl	NULL
tp_ContentTypes	<ContentTypeRef ID="0x0104004475039B7ED69249ADDE60E6DAC4CE4A"><Folder TargetName="Announcement"/></ContentTypeRef><ContentTypeRef ID="0x012000B599492C8570004DACBD86A4B77EAEC9"/><ContentType ID="0x0104"/><ContentType ID="0x0120"/>
Acl	0xF3FE00000100000000000000003000000030000000FFFFFFFFFFFFFFF7F0400000061104308B000000005000000EF12433CB0010000
tp_MaxMajorVersionCount	NULL
tp_MaxMajorwithMinorVersionCount	NULL
tp_SendToLocation	NULL

- [List Web Parts result set](#), specified in section 3.1.4.25.2, which contains all Web Parts defined on the list, one per each row. In this sample, four Web Parts will be returned. The first row is listed as the following:

Column Name	Description
tp_ID	FD655044-665A-4468-BF7D-5CD1B41E1A3A
tp_ListID	D825D4D3-1352-49BB-A498-8DC725D7458F
tp_Type	0

Column Name	Description
tp_Flags	25165825
tp_Hidden	0
tp_Threaded	0
tp_FPModified	0
tp_ReadOnly	0
tp_Scope	[Empty String]
tp_RecurrenceRowset	0
tp_ModerationType	[Empty String]
tp_Personal	0
tp_Ordered	0
tp_DisplayName	All Items
tp_ContentTypeId	0x
tp_PageUrl	/subweb/Lists/MyList/AllItems.aspx
tp_BaseViewId	1
tp_View	NULL
tp_Level	1
tp_IsDefault	0
tp_ZoneID	Main
tp_IsIncluded	0
tp_PartOrder	1
tp_FrameState	0
Id	4905FA83-DEB9-4749-BE22-5E872CCF0521
OnQuickLaunch	1
tp_MobileView	1
tp_MobileDefaultView	1
tp_ContentTypeDefault	0
tp_LockWeb	0
tp_FailIfEmpty	0
tp_FreeForm	0
tp_FileDialog	0

Column Name	Description
tp_AggregationView	0
tp_IncludeRootFolder	0
tp_IncludeVersions	0

- [Event Receivers result set](#), which contains 0 rows because this list does not have any event receiver associated with it.
- [Deleted Fields result set](#), specified in section 3.1.4.25.4, is not returned because this is not an Incremental Export Operation.
- [Null List result set](#), specified in section 3.1.4.25.5, which contains 1 row.

Column Name	Description
ListId	NULL

- Call [proc_DeplGetListItemData](#), specified in section 3.1.4.29, to retrieve the metadata of the list items in the "MyList" list. The stored procedure is called with the following parameters:

Parameter Name	Parameter Value
@SiteId	54F720B1-EC52-44CB-B347-6747DA861CBC
@ LastOrdinal	0
@ VersionSetting	4
@ListId	D825D4D3-1352-49BB-A498-8DC725D7458F
@Columns	nvarchar1,ntext1,nvarchar2,ntext2,datetime1

This stored procedure returns nine result sets.

[List Item Data result set](#), specified in section 3.1.4.29.1, which contains meta data of the list items, one per row. The "MyList" list contains only 1 list item. Therefore, this result set contains only 1 row, as following:

Column Name	Description
Id	6F7A8954-EC20-4861-864A-A634BCA5028D
ExportObjectUrl	subweb/Lists/MyList/1_.000
Ordinal	11
ParentId	D825D4D3-1352-49BB-A498-8DC725D7458F
DirName	subweb/Lists/MyList
LeafName	1_.000
IntId	1
tp_RowOrdinal	0
tp_UIVersion	512
tp_Author	1
tp_Editor	1
tp_Modified	2008-03-03 18:49:33.000
tp_Created	2008-03-03 18:49:33.000
tp_Ordering	NULL
tp_HasAttachment	0
tp_ModerationStatus	0
tp_IsCurrent	1
tp_ItemOrder	100.0
tp_InstanceID	NULL
tp_CopySource	NULL
tp_HasCopyDestinations	NULL
tp_AuditFlags	NULL
tp_InheritAuditFlags	NULL
tp_ContentType	Announcement
tp_ContentTypeId	0x0104004475039B7ED69249ADDE60E6DAC4CE4A
tp_ThreadIndex	NULL
tp_UIVersionString	1.0

Column Name	Description
Id	D2DB5C42-1FB1-4B47-BE2D-85FC7B0D33AC
Type	0
FolderId	F78F95EA-59FB-43A6-860A-8F2F34C06F74
Acl	0xF3FE000001000000000000003000000030000000FFFFFFFFFFFFFFF7F0400000061104308B000000005000000EF12433CB0010000
UserLoginName	NULL
GroupName	NULL
tp_Level	1

- [Field Data result set](#), specified in section 3.1.4.29.2, which contains the field data for the current version of the list item. In this example, the result set contains 1 row, as the following:

Column Name	Description
Id	6F7A8954-EC20-4861-864A-A634BCA5028D
IntId	1
Id	D2DB5C42-1FB1-4B47-BE2D-85FC7B0D33AC
tp_UIVersion	512
tp_UIVersionString	1.0
tp_RowOrdinal	0
nvarchar1	A new announcement!
ntext1	NULL
nvarchar2	NULL
ntext2	<div class=ExternalClass54E2F257DBEA44C0B29B6FC9D1E6F200><div></div></div>
datetime1	NULL
tp_Level	1

- [Multi Lookup result set](#), specified in section 3.1.4.29.3, does not contain any rows because this list does not have any lookup fields to another list.
- [List Item Versions result set](#), specified in section 3.1.4.29.4, contains 1 row because the list item in this list has only 1 version.

Column Name	Description
Id	6F7A8954-EC20-4861-864A-A634BCA5028D
ExportObjectUrl	subweb/Lists/MyList/1_.000

Column Name	Description
Ordinal	11
ParentId	D825D4D3-1352-49BB-A498-8DC725D7458F
DirName	subweb/Lists/MyList
LeafName	1_.000
IntId	1
tp_RowOrdinal	0
tp_UIVersion	512
tp_Author	1
tp_Editor	1
tp_Modified	2008-03-03 18:49:33.000
tp_Created	2008-03-03 18:49:33.000
tp_Ordering	NULL
tp_HasAttachment	0
tp_ModerationStatus	0
tp_IsCurrent	1
tp_ItemOrder	100.0
tp_InstanceID	NULL
tp_CopySource	NULL
tp_HasCopyDestinations	NULL
tp_AuditFlags	NULL
tp_InheritAuditFlags	NULL
tp_ContentType	Announcement
tp_ContentTypeId	0x0104004475039B7ED69249ADDE60E6DAC4CE4A
{tp_ThreadIndex}	NULL
tp_UIVersionString	1.0
{Id}	NULL
{Type}	NULL
{Acl}	NULL
{UserLoginName}	NULL
{GroupName}	NULL

Column Name	Description
tp_Level	1

- [Versions Fields Data result set](#), specified in section 3.1.4.29.5, contains 1 row because the list item in this list has only 1 version.

Column Name	Description
Id	6F7A8954-EC20-4861-864A-A634BCA5028D
IntId	1
DocId	D2DB5C42-1FB1-4B47-BE2D-85FC7B0D33AC
tp_UIVersion	512
tp_UIVersionString	1.0
tp_RowOrdinal	0
nvarchar1	A new announcement!
nvarchar2	NULL
nvarchar3	NULL
ntext1	NULL
ntext2	<div class=ExternalClass54E2F257DBEA44C0B29B6FC9D1E6F200><div></div></div>
datetime1	NULL
tp_Level	1

- [Versions Multi Lookup result set](#), specified by section 3.1.4.29.6, does not contain any rows, because the list does not have any look up fields to another list.
- [List Item Links result set](#), specified by section 3.1.4.29.7, does not contain any rows because the list item in this list does not contain any links.
- [Attachments result set](#), specified in section 3.1.4.29.8, does not contain any rows because the list item in this list does not have any attachments.
- [List Item Event Receivers result set](#), specified in section 3.1.4.29.10, does not contain any rows because there are no event receivers associated with the list item in this list.
- All the objects returned from the previous batch have been processed. Call `proc_DeplGetNextObjectBatch`, specified in section 3.1.4.32, again to get the next batch, with parameter `@LastOrdinal = 11`. The result set does not contain any rows because there are no more data objects that need to be exported.

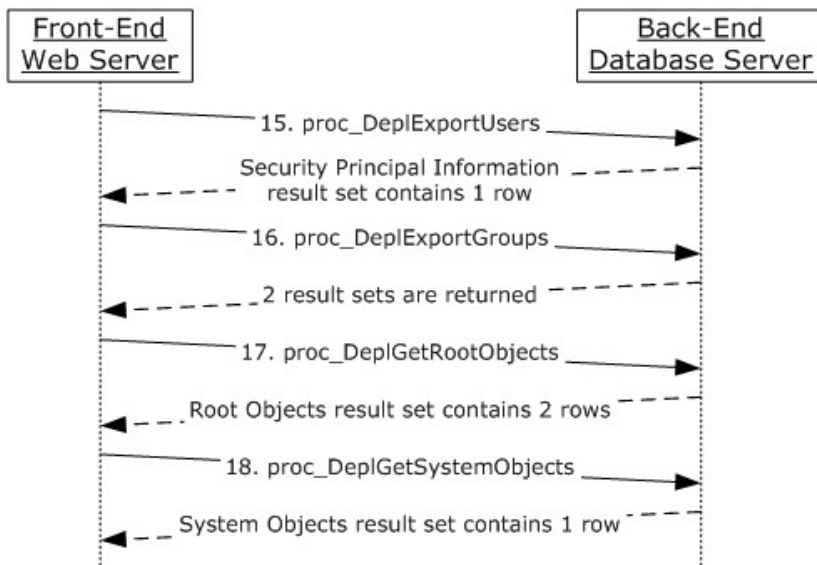


Figure 8: Process used to export a list (Part 3)

- Export security principals. Call [proc_DeplExportUsers](#), specified in section 3.1.4.14, with the following parameters:

Parameter Name	Parameter Value
@SiteId	54F720B1-EC52-44CB-B347-6747DA861CBC
@UserIdCur	-1

The [Security Principal Information result set](#), specified in section 3.1.4.14.1, contains 1 row:

Column Name	Description
tp_SiteId	54F720B1-EC52-44CB-B347-6747DA861CBC
tp_Id	1
tp_Title	User
tp_Login	Redmond\user
tp_Email	user@exchange.microsoft.com
tp_Notes	[Empty String]
tp_DomainGroup	0

Column Name	Description
tp_SiteAdmin	1
tp_SystemId	0x010500000000000515000000A065CF7E784B9B5FE77C87704AB10000
tp_Deleted	0

- Export security groups. Call [proc_DeplExportGroups](#), specified in section 3.1.4.8, with the following parameters

Parameter Name	Parameter Value
@SiteId	54F720B1-EC52-44CB-B347-6747DA861CBC
@GroupIdCur	-1

- [Security Group Information result set](#), specified in section 3.1.4.8.1, contains 1 row as the following:

Column Name	Description
ID	3
Title	Team Site Owners
Description	Use this group to give people full control permissions to the SharePoint site: Team Site
Owner	3
OwnerIsUser	0
RequestEmail	user@ exchange.microsoft.com
Flags	1

- [Security Group Memberships result set](#), specified in section 3.1.4.8.2, contains 1 row as the following:

Column Name	Description
MemberId	1

- Retrieve information about root level objects by calling [proc_DeplGetRootObjects](#), specified in section 3.1.4.36. The [Root Objects result set](#), specified in section 3.1.4.36.1, contains two rows. The first row is shown as the following:

Column Name	Description
Id	F78F95EA-59FB-43A6-860A-8F2F34C06F74

Column Name	Description
Type	2
ParentId	36583090-D164-418C-A626-689B83092AF1
FullUrl	subweb
FullUrl	subweb/Lists/MyList
IsDependency	0
IsSelected	0

- Retrieve information about migration system objects by calling [proc_DeplGetSystemObjects](#), specified in section 3.1.4.37, with @SiteId = '54F720B1-EC52-44CB-B347-6747DA861CBC'. The result set returned one row:

Column Name	Description
Id	F0679A07-C519-4570-A7D9-FF6343187332
Type	3
{Url}	/_catalogs/users

- The client now has collected enough information about the objects to be exported. It serializes this information into the XML file format as described in Deployment Package.

4.2 Import a List

This example shows the communication between the client and the server when importing a Deployment Package which contains one list named "MyList", and one list item inside it. The Deployment Package also contains several other data objects that are necessary for the list to function properly. Such Deployment Package could be created by using this protocol with steps outlined in the previous example.

The import operation will use a number of stored procedures from outside of this protocol. The definition of those stored procedures can be found in [\[MS-WSSFO\]](#) and [\[MS-WSSDLIM\]](#).

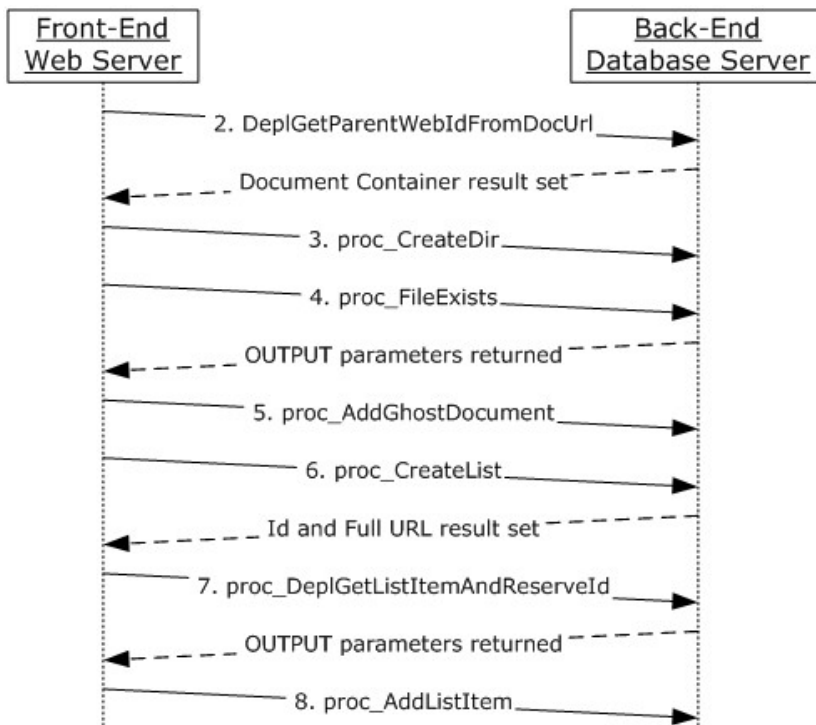


Figure 9: Importing a list

- Prepare the data to be imported. If the data was from a XML package such as a Deployment package, the client is required to parse it and convert the XML file into data that are usable by the stored procedures. In this example, the client has the following data to import. The data will be imported into a Site at <http://server2/sites/import>.

Date Type	Description
List	Name = "My List"
List Item	Title = "A new announcement!"
Folder	Lists/MyList, List support folder
Folder	Lists/MyList/Announcement, List support folder

Date Type	Description
Folder	Lists/MyList/Attachments, List support folder
File	Lists/MyList/AllItems.aspx, List support file
File	Lists/MyList/EditForm.aspx, List support file
File	Lists/MyList/DispForm.aspx, List support file
File	Lists/MyList/NewForm.aspx, List support file

- Call [proc_DeplGetParentWebIdFromDocUrl](#), specified in section 3.1.4.35, to obtain the site and list information of a folder 'Lists/MyList'. The following parameters are used.

Parameter Name	Parameter Value
@SiteId	54F720B1-EC52-44CB-B347-6747DA861CBC
@DocUrl	Sites/Import

The [Location result set](#), specified in section 3.1.4.35.1, returned is the following, which provides the Id of the site where data are being imported to. Many subsequent stored procedure calls will use this information.

Column Name	Description
WebId	8F7BFFAC-5FDA-4B90-B130-DF1CBF13252A
ListId	NULL

- Call `proc_CreateDir`, specified in [MS-WSSFO] section 3.1.4.6, to import the folder "Lists/MyList". The parameters are the following:

Parameter Name	Parameter Value
@DirSiteId	94CC3683-3875-43D0-8DC2-9F82A7D43EB7
@DirWebId	8F7BFFAC-5FDA-4B90-B130-DF1CBF13252A
@DirDirName	Sites/Import
@DirLeafName	MyList
@DirLevel	1
@AddMinorVersion	0
@DocFlags	0
@CreateDirFlags	24
@UserId	1

This stored procedure returns 0 when it creates the folder successfully.

Repeat step 2 and 3 until all folders are imported.

Call [proc_FileExists](#), specified in section 3.1.4.41, to check a file "/importweb/Lists/MyList/AllItems.aspx" already exists before importing it. The parameters are the following:

Parameter Name	Parameter Value
@SiteId	94CC3683-3875-43D0-8DC2-9F82A7D43EB7

Parameter Name	Parameter Value
@WebId	8F7BFFAC-5FDA-4B90-B130-DF1CBF13252A
@DocId	NULL
@Url	sites/import/MyList/AllItems.aspx

The stored procedure returns 0 when it finishes execution successfully. The output parameters are returned as the following:

Parameter Name	Parameter Value
@IsDeleted	0
@DocExists	0
@DestDocId	NULL

The result indicates the file does not exist.

Call [\[MS-WSSDLIM\]](#) section 3.1.4.5 proc_AddGhostDocument to import File "Lists/MyList/AllItems.aspx". The parameters are the following:

Parameter Name	Parameter Value
@SiteId	94CC3683-3875-43D0-8DC2-9F82A7D43EB7
@WebId	8F7BFFAC-5FDA-4B90-B130-DF1CBF13252A
@DocId	1D37ECF4-D2E1-4A98-A440-4A911F4F1E68

Parameter Name	Parameter Value
@DocDirName	sites/import/MyList
@DocLeafName	AllItems.aspx
@Level	1
@UIVersion	512
@EnableMinorVersions	0
@DocSize	2661
@DocFlags	268
@OnRestore	1
@Overwrite	0
@UserId	1
@HasDeleteListItemsRight	0
@SetupPathVersion	3

Parameter Name	Parameter Value
@SetupPath	pages\viewpage.aspx
@SetupPathUser	Redmond\user
@ListId	NULL
@DoclibRowId	NULL
@fCheckQuotaAndWriteLock	0

The store procedure return 0 after the document was successfully imported.

Repeat step 4 and 5 until all files are imported.

Call **proc_CreateList**, specified in [\[MS-WSSDLIM\]](#) section 3.1.4.14, to create the "MyList" list. The parameters are the following:

Parameter Name	Parameter Value
@SiteId	94CC3683-3875-43D0-8DC2-9F82A7D43EB7
@WebId	8F7BFFAC-5FDA-4B90-B130-DF1CBF13252A
@ListId	494CFA0C-BAA8-45F1-9C7B-C8C843DC1C93
@DirName	NULL
@FolderNameBase	NULL

Parameter Name	Parameter Value
@bAlternateUrlOnCollision	1
@Title	MyList
@Version	1
@Author	1
@BaseType	0
@bCreateAttachmentsSubFolder	0
@FeatureId	00BFEA71-D1CE-42DE-9C63-A44004CE0104
@ServerTemplate	104
@DocLibTemplate	NULL
@ImageUrl	/_layouts/images/itann.gif
@ReadSecurity	1
@WriteSecurity	1
@Description	[Empty String]

Parameter Name	Parameter Value
@MajorVersionCount	NULL
@MinorVersionCount	NULL
@Fields ntext	12.0.0.6304.0.0<FieldRef Name="ContentTypeId"/><FieldRef Name="Title" ColName="nvarchar1"/><FieldRef Name="_ModerationComments" ColName="ntext1"/><FieldRef Name="File_x0020_Type" ColName="nvarchar2"/><FieldRef Name="Body" ColName="ntext2"/><FieldRef Name="Expires" ColName="datetime1"/>
@Direction	0
@Flags	545263616
@ThumbnailSize	NULL
@WebImageWidth	NULL
@WebImageHeight	NULL
@bParentFolderChecked	0
@OnRestore	0
@EventSinkAssembly	NULL

Parameter Name	Parameter Value
@EventSinkClass	NULL
@EventSinkData	NULL
@ContentTypes	<ContentTypeRef ID="0x0104004475039B7ED69249ADDE60E6DAC4CE4A"><Folder TargetName="Announcement"/></ContentTypeRef><ContentTypeRef ID="0x012000B599492C8570004DACBD86A4B77EAEC9"/><ContentType ID="0x0104"/><ContentType ID="0x0120"/>
@RootFolderId	8570D03E-2C7D-4863-BABA-B8DC2F01A6D6

Parameter Name	Parameter Value
@SiteId	94CC3683-3875-43D0-8DC2-9F82A7D43EB7
@WebId	8F7BFFAC-5FDA-4B90-B130-DF1CBF13252A
@ListID	494CFA0C-BAA8-45F1-9C7B-C8C843DC1C93
@RowOrdinal	0
@ItemDocType	0
@RetainId	689BE3B3-ABC4-4B81-9F44-FADAE3FDBF02
@RetainObjectIdentity	1
@ItemId	1
@ItemDirName	sites/import/MyList
@ItemLeafName	NULL
@UserID	1
@TimeNow	20080305 00:02:48
@ServerTemplate	104
@Basetype	0
@tp_GUID	4EBEAC61-BDFF-47A5-93B2-52B3F608613C

Parameter Name	Parameter Value
@Level	1
@OnRestore	1
@AddNamespace	1
@CheckDiskQuota	1
@tp_ContentTypeId	0x0104004475039B7ED69249ADDE60E6DAC4CE4A
@nvarchar1	A new announcement!
@ntext1	NULL
@nvarchar2	NULL
@ntext2	<div class=ExternalClass54E2F257DBEA44C0B29B6FC9D1E6F200><div></div></div>
@datetime1	NULL
@tp_ContentType	Announcement
@tp_ModerationStatus	0
@Size	212
@ExtraItemSize	0
@ItemName	A new announcement!
@acl	0xF3FE000001000000000000001000000010000000FFFFFFFFFFFFFFFF7F

The stored procedure returns 0 when the list is created successfully. It also returns the Id and Full Url result set, specified in [\[MS-WSSDLIM\]](#) section 3.1.4.14.2, with 1 row:

Column Name	Description
{ListId}	494CFA0C-BAA8-45F1-9C7B-C8C843DC1C93
{FolderFullUrl}	sites/import/MyList

- Call [proc DepIGetListItemAndReserveId](#), specified in section 3.1.4.26, in preparation to import the list items. The parameters are the following:

Parameter Name	Parameter Value
@SiteId	94CC3683-3875-43D0-8DC2-9F82A7D43EB7

Parameter Name	Parameter Value
@ListId	494CFA0C-BAA8-45F1-9C7B-C8C843DC1C93
@DocId	D2DB5C42-1FB1-4B47-BE2D-85FC7B0D33AC
@WebId	8F7BFFAC-5FDA-4B90-B130-DF1CBF13252A
@OriginalListItemId	1

The stored procedure returns 0 upon completion. It also returns the output parameters as the following:

Parameter Name	Parameter Value
@DocId	D2DB5C42-1FB1-4B47-BE2D-85FC7B0D33AC
@IsDeleted	0
@ListItemId	1
@IsIdTaken	0
@ListItemUrl	NULL

- Call [\[MS-WSSFO\]](#) section 3.1.5.4 proc_AddListItem to import the list items. The parameters are the following:

Parameter Name	Parameter Value
@SiteId	94CC3683-3875-43D0-8DC2-9F82A7D43EB7
@WebId	8F7BFFAC-5FDA-4B90-B130-DF1CBF13252A
@ListID	494CFA0C-BAA8-45F1-9C7B-C8C843DC1C93

Parameter Name	Parameter Value
@RowOrdinal	0
@ItemDocType	0
@RetainId	689BE3B3-ABC4-4B81-9F44-FADAE3FDBF02
@RetainObjectIdentity	1
@UserID	1
@TimeNow	'20080305 00:02:48
@ServerTemplate	104
@Basetype	0
@Level	1
@tp_GUID	"4EBEAC61-BDFF-47A5-93B2-52B3F608613C",
@OnRestore	1,
@AddNamespace	1,
@CheckDiskQuota	1,
@tp_ContentTypeId	0x0104004475039B7ED69249ADDE60E6DAC4CE4A
@nvarchar1	A new announcement!
@ntext1	NULL
@nvarchar2	NULL
@ntext2	<div class=ExternalClass54E2F257DBEA44C0B29B6FC9D1E6F200><div></div></div>
@datetime1	NULL
@tp_ContentType	Announcement
@tp_ModerationStatus	0
@Size	212
@ExtraItemSize	0
@ItemName	A new announcement!
@acl	0xF3FE000001000000000000001000000010000000FFFFFFFFFFFFFF7F

This stored procedure returns 0 after the list items is successfully created.

4.3 Remove Unused Web Parts After Importing a File

This example shows queries that are used to directly update the [WebParts Table](#) and [WebPartLists Table](#), sections 2.2.6.1 and 2.2.6.2, to find and remove unused web parts after a file is imported. The steps match the ones shown in 3.2.4.4

1. Create the [#WebParts Table](#)

```
CREATE TABLE #WebParts (WebPartId uniqueidentifier NOT NULL PRIMARY KEY)
```

2. While importing the file, add the web parts referenced by this file to the #WebParts Table

```
INSERT INTO #WebParts VALUES(@WebPartId) 'BA8CB0D4-CD11-43F7-823E-0394DFFAF891'
```

3. Delete any rows in the WebParts Table and WebPartLists Table, sections 2.2.6.1 and 2.2.6.2, if the web part represented by that row is associated with the file being imported, but the identifier for the Web Part is not contained in the #WebParts Table.

```
DELETE WebPartLists FROM
  (SELECT WP2.tp_ID FROM WebParts AS WP2 WHERE
    WP2.tp_SiteId = @SiteId AND
    WP2.tp_PageUrlID = @DocId AND
    WP2.tp_UserID IS NULL AND
    (NOT EXISTS (SELECT 1 FROM #WebParts AS TempWP WHERE TempWP.WebPartId =
WP2.tp_ID))
  ) AS WP1
WHERE
  WebPartLists.tp_SiteId = @SiteId AND
  WebPartLists.tp_WebPartID = WP1.tp_ID AND (WebPartLists.tp_Level = 1 )

DELETE WebParts FROM
  (SELECT WP2.tp_ID FROM WebParts AS WP2 WHERE
    WP2.tp_SiteId = @SiteId AND
    WP2.tp_PageUrlID = @DocId AND
    WP2.tp_UserID IS NULL AND
    (NOT EXISTS (SELECT 1 FROM #WebParts AS TempWP WHERE TempWP.WebPartId =
WP2.tp_ID))
  ) AS WP1
WHERE
  WebParts.tp_SiteId = @SiteId AND
  WebParts.tp_ID = WP1.tp_ID AND (WebParts.tp_Level = 1 )
```

Parameter Name	Parameter Value	Description
@SiteId	7537FBDF-2B82-43E7-9F16-9251CA8879FE	Identifier of the Site Collection which contains the document being imported.
@DocId	CEED281B-7684-43C4-A6B4-6A25DA1558EB	Identifier of the document that is being imported

- Clean up all the rows in the #WebParts Table

```
TRUNCATE TABLE #WebParts
```

5 Security

5.1 Security Considerations for Implementers

Interactions with SQL are susceptible to tampering and other forms of security risks. Implementers are advised to sanitize input parameters for stored procedures prior to invoking the stored procedure.

5.2 Index of Security Parameters

Security Parameter	Section
proc_DeplExportGroups	3.1.4.8
proc_DeplExportGroupX	3.1.4.9
proc_DeplExportRoleAssignments	3.1.4.10
proc_DeplExportRoleAssignmentX	3.1.4.11
proc_DeplExportRoles	3.1.4.12
proc_DeplExportRoleX	3.1.4.13
proc_DeplExportUsers	3.1.4.14
proc_DeplExportUserX	3.1.4.15

6 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- Microsoft® SQL Server® 2005
- Microsoft® SQL Server® 2008
- Microsoft® SQL Server® 2008 R2
- Windows® SharePoint® Services 3.0

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

[<1> Section 3.1.4.1.1:](#) Windows SharePoint Services 3.0 Service Pack 1 (SP1) returns 87 when @id is empty and @type = 4.

[<2> Section 3.1.4.10.2:](#) In Windows SharePoint Services 3.0 SP1, tp_Id is not NULL when @WssGroupOnly is true.

[<3> Section 3.1.4.22.7:](#) Prior to the Infrastructure Update for Windows SharePoint Services 3.0 (KB951695), the Event Receivers Result Set returns one row for each Event Receiver that is registered for the first 200 files to be exported with ordinals greater than @LastOrdinal. As of the Infrastructure Update for Windows SharePoint Services 3.0 (KB951695), the Event Receivers Result Set returns zero rows.

7 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

8 Index

[#WebParts table](#) 16

A

Abstract data model

[client](#) 83

[server](#) 18

[Applicability](#) 12

[Attribute groups - overview](#) 17

[Attributes - overview](#) 17

B

[Binary structures - overview](#) 15

C

[Capability negotiation](#) 12

[Change tracking](#) 127

Client

[abstract data model](#) 83

[details](#) 82

[initialization](#) 83

[local events](#) 87

[message processing](#) 83

[sequencing rules](#) 83

[timer events](#) 87

[timers](#) 83

Common data types

[overview](#) 13

[Complex types - overview](#) 16

D

Data model - abstract

[client](#) 83

[server](#) 18

Data types

[common](#) 13

[Web Part moderation simple type](#) 13

[Web Part scope simple type](#) 13

Data types – simple

[Web Part moderation](#) 13

[Web Part scope](#) 13

[Deployment object type enumeration](#) 14

Details

[client](#) 82

E

[Elements - overview](#) 16

Enumerations

[deployment object type](#) 14

[role type](#) 15

Events

[local - client](#) 87

[local - server](#) 82

[timer - client](#) 87

[timer - server](#) 82

Examples

[export a list](#) 88

[import a list](#) 110

[remove unused Web Parts after importing a file](#) 122

[Export a list example](#) 88

[Export sequencing rules](#) 83

F

[Fields - vendor-extensible](#) 12

Flag structures

[security group membership](#) 13

G

[Glossary](#) 8

[Groups - overview](#) 17

I

[Implementer - security considerations](#) 125

[Import a list example](#) 110

[Incremental export sequencing rules](#) 85

[Index of security parameters](#) 125

[Informative references](#) 11

Initialization

[client](#) 83

[server](#) 18

[Introduction](#) 8

L

[Link documents export sequencing rules](#) 85

Local events

[client](#) 87

[server](#) 82

M

Message processing

[client](#) 83

[server](#) 18

Messages

[attribute groups](#) 17

[attributes](#) 17

[binary structures](#) 15

[common data types](#) 13

[complex types](#) 16

[elements](#) 16

[groups](#) 17

[namespaces](#) 16

[result sets](#) 15

[security group membership flag structure](#) 13

[simple types](#) 16

[transport](#) 13

[XML structures](#) 16

Methods

[proc_DeplAddExportObject](#) 18

[proc_DeplAddExportObjectDependencies](#) 20
[proc_DeplAddExportObjectLinks](#) 21
[proc_DeplAddIncrementalSearchScope](#) 22
[proc_DeplAddSite](#) 22
[proc_DeplCalculateChildrenToExport](#) 23
[proc_DeplCalculateIncrementalSearchScope](#) 24
[proc_DeplExportGroups](#) 24
[proc_DeplExportGroupX](#) 26
[proc_DeplExportRoleAssignments](#) 27
[proc_DeplExportRoleAssignmentX](#) 29
[proc_DeplExportRoles](#) 30
[proc_DeplExportRoleX](#) 31
[proc_DeplExportUsers](#) 33
[proc_DeplExportUserX](#) 34
[proc_DeplFindScopeObject](#) 35
[proc_DeplFixWebPartsOnListCreation](#) 36
[proc_DeplGetContentTypeData](#) 37
[proc_DeplGetContentTypesToParse](#) 37
[proc_DeplGetDocLibRootFolderUrl](#) 38
[proc_DeplGetFileData](#) 39
[proc_DeplGetFolderById](#) 48
[proc_DeplGetFolderData](#) 49
[proc_DeplGetIncrementalChanges](#) 51
[proc_DeplGetListData](#) 52
[proc_DeplGetListItemAndReserveId](#) 60
[proc_DeplGetListItemByDocId](#) 61
[proc_DeplGetListItemByUrl](#) 62
[proc_DeplGetListItemData](#) 62
[proc_DeplGetLookupItemData](#) 74
[proc_DeplGetLookupListUrl](#) 75
[proc_DeplGetNextObjectBatch](#) 75
[proc_DeplGetParentWebIdForList](#) 76
[proc_DeplGetParentWebIdFromDocId](#) 77
[proc_DeplGetParentWebIdFromDocUrl](#) 77
[proc_DeplGetRootObjects](#) 78
[proc_DeplGetSystemObjects](#) 79
[proc_DeplOrderExportObjects](#) 79
[proc_DeplReserveNextAvailableId](#) 80
[proc_DeplUpdateWebProperties](#) 80
[proc_FileExists](#) 81

N

[Namespaces](#) 16
[Normative references](#) 10

O

[Overview \(synopsis\)](#) 11

P

[Parameters - security index](#) 125
[Preconditions](#) 11
[Prerequisites](#) 11
[proc_DeplAddExportObject](#) method 18
[proc_DeplAddExportObjectDependencies](#) method 20
[proc_DeplAddExportObjectLinks](#) method 21
[proc_DeplAddIncrementalSearchScope](#) method 22
[proc_DeplAddSite](#) method 22
[proc_DeplCalculateChildrenToExport](#) method 23

[proc_DeplCalculateIncrementalSearchScope](#) method 24
[proc_DeplExportGroups](#) method 24
[proc_DeplExportGroupX](#) method 26
[proc_DeplExportRoleAssignments](#) method 27
[proc_DeplExportRoleAssignmentX](#) method 29
[proc_DeplExportRoles](#) method 30
[proc_DeplExportRoleX](#) method 31
[proc_DeplExportUsers](#) method 33
[proc_DeplExportUserX](#) method 34
[proc_DeplFindScopeObject](#) method 35
[proc_DeplFixWebPartsOnListCreation](#) method 36
[proc_DeplGetContentTypeData](#) method 37
[proc_DeplGetContentTypesToParse](#) method 37
[proc_DeplGetDocLibRootFolderUrl](#) method 38
[proc_DeplGetFileData](#) method 39
[proc_DeplGetFolderById](#) method 48
[proc_DeplGetFolderData](#) method 49
[proc_DeplGetIncrementalChanges](#) method 51
[proc_DeplGetListData](#) method 52
[proc_DeplGetListItemAndReserveId](#) method 60
[proc_DeplGetListItemByDocId](#) method 61
[proc_DeplGetListItemByUrl](#) method 62
[proc_DeplGetListItemData](#) method 62
[proc_DeplGetLookupItemData](#) method 74
[proc_DeplGetLookupListUrl](#) method 75
[proc_DeplGetNextObjectBatch](#) method 75
[proc_DeplGetParentWebIdForList](#) method 76
[proc_DeplGetParentWebIdFromDocId](#) method 77
[proc_DeplGetParentWebIdFromDocUrl](#) method 77
[proc_DeplGetRootObjects](#) method 78
[proc_DeplGetSystemObjects](#) method 79
[proc_DeplOrderExportObjects](#) method 79
[proc_DeplReserveNextAvailableId](#) method 80
[proc_DeplUpdateWebProperties](#) method 80
[proc_FileExists](#) method 81
[Product behavior](#) 126

R

[References](#) 10
 [informative](#) 11
 [normative](#) 10
[Relationship to other protocols](#) 11
[Remove unused Web Parts after importing a file example](#) 122
[Result sets - overview](#) 15
[Role type enumeration](#) 15

S

Security
 [implementer considerations](#) 125
 [parameter index](#) 125
 [Security group membership flag structure](#) 13
 Sequencing rules
 [client](#) 83
 [incremental export](#) 85
 [link documents export](#) 85
 [server](#) 18
 [unused Web Parts removal](#) 86
 Server

- [abstract data model](#) 18
- [initialization](#) 18
- [local events](#) 82
- [message processing](#) 18
- [proc_DeplAddExportObject method](#) 18
- [proc_DeplAddExportObjectDependencies method](#) 20
- [proc_DeplAddExportObjectLinks method](#) 21
- [proc_DeplAddIncrementalSearchScope method](#) 22
- [proc_DeplAddSite method](#) 22
- [proc_DeplCalculateChildrenToExport method](#) 23
- [proc_DeplCalculateIncrementalSearchScope method](#) 24
- [proc_DeplExportGroups method](#) 24
- [proc_DeplExportGroupX method](#) 26
- [proc_DeplExportRoleAssignments method](#) 27
- [proc_DeplExportRoleAssignmentX method](#) 29
- [proc_DeplExportRoles method](#) 30
- [proc_DeplExportRoleX method](#) 31
- [proc_DeplExportUsers method](#) 33
- [proc_DeplExportUserX method](#) 34
- [proc_DeplFindScopeObject method](#) 35
- [proc_DeplFixWebPartsOnListCreation method](#) 36
- [proc_DeplGetContentTypeData method](#) 37
- [proc_DeplGetContentTypesToParse method](#) 37
- [proc_DeplGetDocLibRootFolderUrl method](#) 38
- [proc_DeplGetFileData method](#) 39
- [proc_DeplGetFolderById method](#) 48
- [proc_DeplGetFolderData method](#) 49
- [proc_DeplGetIncrementalChanges method](#) 51
- [proc_DeplGetListData method](#) 52
- [proc_DeplGetListItemAndReserveId method](#) 60
- [proc_DeplGetListItemByDocId method](#) 61
- [proc_DeplGetListItemByUrl method](#) 62
- [proc_DeplGetListItemData method](#) 62
- [proc_DeplGetLookupItemData method](#) 74
- [proc_DeplGetLookupListUrl method](#) 75
- [proc_DeplGetNextObjectBatch method](#) 75
- [proc_DeplGetParentWebIdForList method](#) 76
- [proc_DeplGetParentWebIdFromDocId method](#) 77
- [proc_DeplGetParentWebIdFromDocUrl method](#) 77
- [proc_DeplGetRootObjects method](#) 78
- [proc_DeplGetSystemObjects method](#) 79
- [proc_DeplOrderExportObjects method](#) 79
- [proc_DeplReserveNextAvailableId method](#) 80
- [proc_DeplUpdateWebProperties method](#) 80
- [proc_FileExists method](#) 81
- [sequencing rules](#) 18
- [timer events](#) 82
- [timers](#) 18
- Simple types
 - [Web Part moderation](#) 13
 - [Web Part scope](#) 13
- [Simple types - overview](#) 16
- [Standards assignments](#) 12
- Structures
 - [binary](#) 15
 - [XML](#) 16

T

- Tables
 - [#WebParts](#) 16
 - [WebPartLists](#) 16
 - [WebParts](#) 15
- Timer events
 - [client](#) 87
 - [server](#) 82
- Timers
 - [client](#) 83
 - [server](#) 18
- [Tracking changes](#) 127
- [Transport](#) 13
- Types
 - [complex](#) 16
 - [simple](#) 16
- U**
 - [Unused Web Parts removal sequencing rules](#) 86
- V**
 - [Vendor-extensible fields](#) 12
 - [Versioning](#) 12
- W**
 - [Web Part moderation simple type](#) 13
 - [Web Part scope simple type](#) 13
 - [WebPartLists table](#) 16
 - [WebParts table](#) 15
- X**
 - [XML structures](#) 16