[MS-WSSPROG]:
Windows SharePoint Services:
Content Database Communications Programmability Extensions

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
12/12/2008	1.01	Editorial	Revised and edited the technical content
03/18/2009	1.02	Editorial	Revised and edited the technical content
07/13/2009	1.03	Major	Changes made for template compliance
08/28/2009	1.04	Editorial	Revised and edited the technical content
11/06/2009	1.05	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.
08/26/2010	2.05	Major	Significantly changed the technical content.
09/27/2010	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
06/10/2011	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
01/20/2012	2.6	Minor	Clarified the meaning of the technical content.
04/11/2012	2.6	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	2.6	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction

This document sepecifies the Windows SharePoint Services: Content Database Programmability Extensions Communications Protocol. This protocol specifies the communication sequences used by front-end Web servers and application servers to perform data query and change commands on back-end database servers as part of Web Part, event receiver, workflow, and work item operations.

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

1.1 Glossary

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

anonymous user
Coordinated Universal Time (UTC)
GUID
object
XML

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

All Users
assembly
assembly name
attachment
author
back-end database server
base view identifier
binary payload
CAML
configuration database
content database
content type
context collection
context object
context type
current user
current version
customizable
Data View Web Part
datetime
declarative workflow association
default list view
default view
delete transaction identifier
deleted
directory name
display name
document
document library
document version
draft

empty GUID
event
event host
event receiver
event receiver source
exponential backoff
feature
folder
front-end Web server
fully qualified class name
item identifier
leaf name
list
List Form Web Part
list identifier
list item
list view
list view page
List View Web Part
minor version control
mobile device
page
permission
personal view
personal Web Part
public view
published
publishing level
query
Recycle Bin
Recycle Bin item
Recycle Bin item list
result set
return code
sequence number
shared view
site
site collection
site collection identifier
stored procedure
store-relative form
Structured Query Language (SQL)
SystemID
text payload
throttled fetch
timer job
Transact-Structured Query Language (T-SQL)
Uniform Resource Locator (URL)
user identifier
version control
view
Web Part
Web Part cache
Web Part chrome state
Web Part Page

Web Part property
Web Part type identifier
Web Part zone
Web Part zone identifier
Web Part zone index
work item
work item batch
work item batch identifier
work item identifier
work item parent identifier
work item process
work item subtype
work item subtype identifier
work item type
work item type identifier
workflow
workflow association
workflow history list
workflow instance
workflow task
workflow task list
workflow template
XML schema

The following terms are specific to this document:

declarative workflow: A workflow that is created with XAML (Extensible Application Markup Language) files and does not require precompiled code to run.

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

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

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

[MS-WSSCADM] Microsoft Corporation, "Windows SharePoint Services Content Database Administrative Communications Protocol Specification".

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

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

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

[XMLSCHEMA1] Thompson, H.S., Ed., Beech, D., Ed., Maloney, M., Ed., and Mendelsohn, N., Ed., "XML Schema Part 1: Structures", W3C Recommendation, May 2001, http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/

[XMLSCHEMA2] Biron, P.V., Ed. and Malhotra, A., Ed., "XML Schema Part 2: Datatypes", W3C Recommendation, May 2001, http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/

1.2.2 Informative References

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

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

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

1.3 Protocol Overview (Synopsis)

This protocol specifies the communication between the **front-end Web server** and the **back-end database server**. This communication satisfies requests associated with **events**, **Web Parts**, **workflows**, and **work items**. This client/server protocol uses the Tabular Data Stream Protocol as its transport between the client, and the back end database server.

1.3.1 Event Operations

The protocol specifies methods for creating, retrieving, manipulating and deleting events. When client requests for event information are sent to the front-end Web server, the front-end Web server sends a series of **stored procedure** calls to the back end database server for the requested information. The stored procedures return data which in turn can be used for further calls to other stored procedures. The front-end Web server turns the values in the **return codes** and **result sets** into the data and metadata for the events requested by the client, and sends it back to the client using the same protocol used by the initial request.

1.3.2 Web Part Operations

The protocol specifies methods for creating, retrieving, manipulating, and deleting Web Parts. When client requests for Web Part information are sent to the front-end Web server, the front-end Web server sends a series of stored procedure calls to the back end database server for the requested information. The stored procedures return data which in turn can be used for further calls to other stored procedures. The front-end Web server turns the values in the return codes and result sets into the data and metadata for the Web Parts requested by the client, and sends it back to the client using the same protocol used by the initial request.

1.3.3 Workflow Operations

The protocol specifies methods for creating, retrieving, manipulating, and deleting workflows. When client requests for workflow information are sent to the front-end Web server, it responds with a series of stored procedure calls to the back end database server for the requested information. The stored procedures return data which in turn can be used for further calls to other stored procedures. The front-end Web server turns the values in the return codes and result sets into the data and metadata for the workflow requested by the client, and sends it back to the client using the same protocol used by the initial request.

1.3.4 Work Item Operations

The protocol specifies methods for creating, retrieving, manipulating and deleting work items. When client requests for work item information are sent to the front-end Web server, it responds with a series of stored procedure calls to the back end database server for the requested information. The stored procedures return data which in turn can be used for further calls to other stored procedures. The front-end Web server turns the values in the return codes and result sets into the data and metadata for the work items requested by the client, and sends it back to the client using the same protocol used by the initial request.

1.4 Relationship to Other Protocols

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

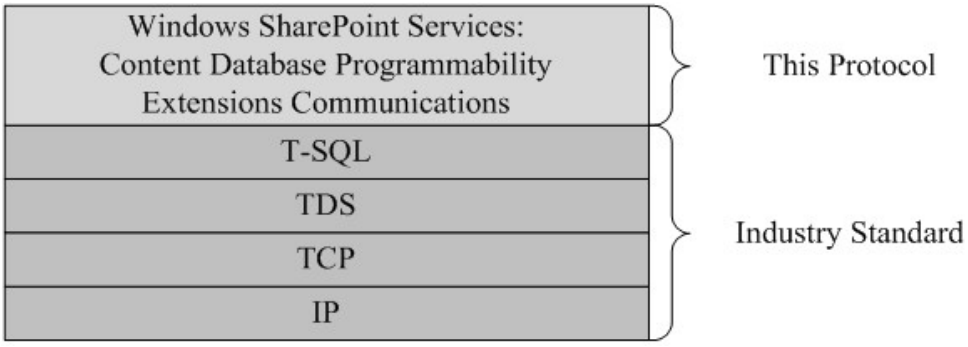

Figure 1: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

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

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

1.6 Applicability Statement

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

1.7 Versioning and Capability Negotiation

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

1.8 Vendor-Extensible Fields

This protocol has the following vendor extensible fields:

workflow instance data – A binary structure that contains the state of a workflow. This binary can be passed into or retrieved by the following sub-protocols and result sets:

- proc_UpdateListItemWorkflowInstanceData
- List Item Workflows Result Set

work item binary payload – An arbitrary binary stored with a work item that can be used by the protocol client that runs the work item. This binary can be passed into or retrieved by the following sub-protocols:

- proc_AddWorkItem
- proc_UpdateWorkItem
- Work Items Result Set

work item text payload – A arbitrary string stored with a work item that can be used by the protocol client that runs the work item. This string can be passed into or retrieved by the following subprotocols:

- proc_AddWorkItem
- proc_UpdateWorkItem
- Work Items Result Set

Workflow Modification Data – XML that contains data about a workflow. See Workflow Modifications for schema information. This XML can be passed into or retrieved by the following sub-protocols and result sets:

- proc_UpdateListItemWorkflowInstanceData
- List Item Workflows Result Set

workflow association data – XML that contains information about a **workflow association**. This XML can be passed into or retrieved by the following sub-protocols and result sets:

- proc_AddWorkflowAssociation
- proc_UpdateWorkflowAssociation
- Workflow Associations Result Set

Vendors are free to choose their own values for these fields. This protocol specifies no mechanism for guaranteeing uniqueness of vendor-specific values for these fields.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

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

2.2 Common Data Types

This section contains common definitions used by this protocol.

2.2.1 Simple Data Types and Enumerations

2.2.2 Simple Data Types

2.2.2.1 Context Collection Identifier

A **GUID** used to identify a **context collection**.

2.2.2.2 Context Identifier

A GUID used to identify an **object** or a group of objects related to an **event receiver**.

2.2.2.3 Context Object Identifier

A GUID used to identify the **context object** for the **event host** for which an Event Receiver is registered.

2.2.2.4 Context Type Identifier

This is a GUID used to identify a **context type**.

2.2.2.5 Event Receiver Source Identifier

This is a GUID used to identify an **event receiver source**.

2.2.2.6 List Item Version

A 4-byte integer counter incremented any time a change is made to the properties of a **list item**, and is used for optimistic concurrency conflict detection.

2.2.2.7 Workflow Template Identifier

A GUID used to identify the **workflow template**.

2.2.3 Bit Fields and Flag Structures

2.2.3.1 Event Receiver Source Type

This is a 4-byte integer that specifies the Event Receiver Source of an Event Receiver. All valid values for this type are specified in the following table.

Value	Meaning
0	No specific Event Receiver Source.
1	The Event Receiver Source is a **content type**.
2	The Event Receiver Source is a **feature**.

2.2.3.2 Workflow Association Configuration

A 32-bit mask describing the configuration of the workflow association. The only valid values of the Workflow association Configuration bits are specified in the following table.

Value	Name	Meaning
0x00000001	WFA_AUTO_START_ADD	The protocol server MUST create and run a workflow whenever a new list item is created in the **list** with which the workflow association is associated .
0x00000002	WFA_AUTO_START_CHANGE	The protocol server MUST create and run a workflow whenever a list item is modified in the list with which the workflow association is associated.
0x00000008	WFA_ALLOW_MANUAL_START	Users are allowed to manually create and run workflows created from the workflow association.
0x00000010	WFA_HAS_STATUS_COLUMN	The workflow association has a workflow status field.
0x00000020	WFA_LOCK_ITEM	When a front-end Web server is processing a workflow created from the workflow association, it MUST lock the workflow.
0x00000040	WFA_DECLARATIVE	The workflow association is a **declarative workflow association**.
0x00000080	WFA_NO_NEWWORKFLOWS	The server MUST NOT create any new workflows from the workflow association.
0x00000200	WFA_MARKED_FOR_DELETE	The workflow association has been marked for deletion by proc_AutoCleanupWorkflows.
0x00001000	WFA_COMPRESS_INSTANCEDATA	The **workflow instance** data of workflows create from the workflow association is compressed.
0x00008000	WFA_ALLOW_ASYNCMANUALSTART	If a workflow created from the workflow association cannot be manually started immediately, it will be scheduled for later processing.

2.2.3.3 Workflow Internal State

A 32-bit mask describing the state of the workflow. The only valid values of the Workflow Internal State bits are specified as follows:

Value	Name	Meaning
0x00000001	WFS_LOCKED	A front-end Web server has locked the workflow for processing.

Value	Name	Meaning
		No other front-end Web server can process the workflow.
0x00000002	WFS_RUNNING	A front-end Web server is processing the workflow.
0x00000004	WFS_COMPLETED	The workflow has completely processed. No further processing can be done.
0x00000008	WFS_CANCELED	The workflow was canceled by a user. No further processing can be done.
0x00000040	WFS_FAULTING	The workflow has encountered a faulting error.
0x00000080	WFS_TERMINATED	The workflow was terminated before being completely processed. No further processing can be done.
0x00000100	WFS_SUSPENDED	The workflow was suspended. The workflow can resume running.
0x00000400	WFS_HASNEWEVENTS	The workflow has events that need to be processed.
0x00000800	WFS_NOTSTARTED	The workflow has not yet started running.
0x00001000	WFS_HASWAKEUPTIME	A work item has been created to resume processing the workflow.

2.2.3.4 Workflow Status1

An integer describing the status of the workflow. The following values are defined, but the field is vendor extensible and other values are allowed:

Value	Name	Description
0	WFSTAT_NOTSTARTED	The workflow has not yet started running.
1	WFSTAT_FAILEDTOSTART	The workflow failed to start.
2	WFSTAT_INPROGRESS	A front-end Web server is processing the workflow.
3	WFSTAT_FAULTING	The workflow has encountered a faulting error.
4	WFSTAT_USERCANCEL	The workflow was canceled by a user.
5	WFSTAT_COMPLETED	The workflow has completely processed.
6	WFSTAT_FAILEDTOSTART_RETRY	The workflow failed to start. Processing can be attempted again.
7	WFSTAT_FAULTING_RETRY	The workflow has encountered a faulting error. Processing can be attempted again.

2.2.4 Enumerations

None.

2.2.5 Binary Structures

None.

2.2.6 Result Sets

2.2.6.1 List Item Workflows Result Set

The List Item Workflows Result Set returns information about workflows created for List items. The **T-SQL** syntax for the result set is as follows:

```
Id                      uniqueidentifier,
TemplateId              uniqueidentifier,
ListId                  uniqueidentifier,
SiteId                  uniqueidentifier,
WebId                   uniqueidentifier,
ItemId                  int,
ItemGUID                uniqueidentifier,
TaskListId              uniqueidentifier,
AdminTaskListId         varbinary(16),
Author                  int,
Modified                datetime,
Created                 datetime,
StatusVersion           int,
Status1                 int,
Status2                 int,
Status3                 int,
Status4                 int,
Status5                 int,
Status6                 int,
Status7                 int,
Status8                 int,
Status9                 int,
Status10                int,
TextStatus1             nvarchar(128),
TextStatus2             nvarchar(128),
TextStatus3             nvarchar(128),
TextStatus4             nvarchar(128),
TextStatus5             nvarchar(128),
Modifications           ntext,
InstanceData            image,
InstanceDataSize        int,
InternalState           int,
ProcessingId            int;
```

Id: The workflow identifier of the workflow. This value MUST NOT be NULL.

TemplateId: The Workflow Template identifier of the workflow Template from which the Workflow was created. This value MUST NOT be NULL.

ListId: The List identifier, as specified in [MS-WSSFO], section 2.2.1.5, of the List containing the List Item for which the workflow was created. This value MUST NOT be NULL.

SiteId: The **site collection identifier** of the **site collection** which contains the workflow. This value MUST NOT be NULL.

WebId: The Site identifier, as specified in [MS-WSSFO], section 2.2.1.10, of the **site** which contains the workflow. This value MUST NOT be NULL.

ItemId: The List Item identifier, as specified in [MS-WSSFO], section 2.2.1.6, of the List Item for which the workflow was created.

ItemGUID: The item GUID of the List Item.

TaskListId: The List identifier, as specified in [MS-WSSFO], section 2.2.1.5, of the **workflow task list** of the workflow.

AdminTaskListId: This column MUST be NULL.

Author: The **user identifier** of the user that created the workflow.

Modified: The date and time in UTC when the workflow was last modified. This value MUST NOT be NULL.

Created: The date and time in UTC when the workflow was created.

StatusVersion: The StatusVersion value for the workflow. This value MUST NOT be NULL.

Status1: TheWorkflow Status1 value for the workflow.

Status2: The protocol client MUST ignore this value.

Status3: The protocol client MUST ignore this value.

Status4: The protocol client MUST ignore this value.

Status5: The protocol client MUST ignore this value.

Status6: The protocol client MUST ignore this value.

Status7: The protocol client MUST ignore this value.

Status8: The protocol client MUST ignore this value.

Status9: The protocol client MUST ignore this value.

Status10: The protocol client MUST ignore this value.

TextStatus1: The protocol client MUST ignore this value.

TextStatus2: The protocol client MUST ignore this value.

TextStatus3: The protocol client MUST ignore this value.

TextStatus4: The protocol client MUST ignore this value.

TextStatus5: The protocol client MUST ignore this value.

Modifications: The Workflow Modifications of the workflow.

InstanceData: The workflow instance data of the workflow.

InstanceDataSize: The size of the instance data in InstanceData. If InstanceData is NULL, this field MUST contain the value 0.

InternalState: The workflow internal state for the workflow.

ProcessingId: The workflow process identifier of the workflow process running the workflow.

2.2.6.2 Web Parts Result Set

Web Parts Result Set returns properties of the Web Parts. There MUST be one row per Web Part in this Result Set. The T-SQL syntax for the result set is as follows:

```
tp_ID                       uniqueidentifier,
tp_ListId                   uniqueidentifier,
tp_Type                     tinyint,
tp_Flags                    int,
tp_DisplayName              nvarchar(255),
tp_Version                  int,
{DocumentUrl}               nvarchar(385),
tp_PartOrder                int,
tp_ZoneID                   nvarchar(64),
tp_IsIncluded               bit,
tp_FrameState               tinyint,
tp_WebPartTypeId            uniqueidentifier,
tp_AllUsersProperties       image,
tp_PerUserProperties        image,
tp_Cache                    image,
tp_Source                   ntext;
```

tp_ID: The Web Part identifier, as specified in [MS-WSSFO], section 2.2.1.14. This value MUST NOT be NULL.

tp_ListId: The List identifier, as specified in [MS-WSSFO], section 2.2.1.5, of the List to which the Web Part refers.

tp_Type: The Page type, as specified in [MS-WSSFO2] section 2.2.3.14, of the **Web Part Page** that contains the Web Part.

tp_Flags: The View Flags, as specified in [MS-WSSFO], section 2.2.2.11, of the Web Part.

tp_DisplayName: The **display name** of the Web Part.

tp_Version: This value MUST be ignored.

{DocumentUrl}: The **store-relative form URL** of the Web Part Page that contains the Web Part. This value MUST NOT be NULL.

tp_PartOrder: The **Web Part zone index** of the Web Part.

tp_ZoneID: The **Web Part zone identifier** of the Web Part.

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_WebPartTypeId: The **Web Part type identifier** of the Web Part.

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

tp_PerUserProperties: A serialized representation of zero or more personalizable properties on the Web Part. If this value is NULL, then default values will be used for all of the personalizable properties on the Web Part.

tp_Cache: Private data cache of the Web Part.

tp_Source: The **Web Part properties** of the Web Part in either WPV2:WebPart format (as specified in [MS-WPPS] section 2.2.4.2) or HTML format.

2.2.6.3 Workflow Associations Result Set

The Workflow Associations Result Set returns Workflow associations, one per row. The T-SQL syntax for the result set is as follows:

```
Id                    uniqueidentifier NOT NULL,
BaseId                uniqueidentifier NOT NULL,
ParentId              varbinary(16),
Name                  nvarchar(255),
Description           nvarchar(1023),
StatusFieldName       nvarchar(64),
SiteId                uniqueidentifier NOT NULL,
WebId                 varbinary(16),
ListId                varbinary(16),
ContentTypeId         varbinary(512),
InstanceCount         int,
TaskListId            varbinary(16),
HistoryListId         varbinary(16),
TaskListTitle         nvarchar(255),
HistoryListTitle      nvarchar(255),
Author                int,
Created               datetime,
Modified              datetime,
PermissionsManual     bigint,
Version               int,
AutoCleanupDays       int,
InstantiationParams   ntext,
Configuration         int;
```

Id: The Workflow association identifier of the workflow association.

BaseId: The Workflow Template identifier of the workflow Template on which the workflow association is based.

ParentId: The workflow association identifier of the parent workflow association of the Workflow association of the row.

Name: The display name of the workflow association.

Description: The description of the workflow association.

StatusFieldName: The display name of the workflow status field of the workflow association.

SiteId: The Site Collection identifier of the Site Collection containing the workflow association.

WebId: The Site identifier (as specified in [MS-WSSFO], section 2.2.1.10) of the Site containing the workflow association.

ListId: The List identifier (as specified in [MS-WSSFO], section 2.2.1.5) of the list with which the workflow association is associated.

ContentTypeId: The Content type identifier (as specified in [MS-WSSFO], section 2.2.1.1) of the Content type with which the Workflow is associated.

InstanceCount: The current number of active workflows created from the workflow association.

TaskListId: The List identifier (as specified in [MS-WSSFO], section 2.2.1.5) of the workflow Task List of the workflow association.

HistoryListId: The List identifier (as specified in [MS-WSSFO], section 2.2.1.5) of the **workflow history list** of the workflow association.

TaskListTitle: The display name of the workflow Task List of the workflow association.

HistoryListTitle: The display name of the workflow History List of the workflow association.

Author: The User identifier of the **author** of the workflow association.

Created: The date and time in UTC when the workflow association was created.

Modified: The date and time in UTC when the workflow association was last modified.

PermissionsManual: The WSS Rights Mask (as specified in [MS-WSSFO2] section 2.2.2.14) required to manually start any workflows created from the workflow association.

Version: The version of the workflow association.

AutoCleanupDays: The number of days after which completed workflows created from the workflow association will be automatically deleted.

InstantiationParams: The workflow association data of the workflow.

Configuration: The Workflow association Configuration value for the Workflow association.

2.2.6.4 Work Items Result Set

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

```
DeliveryDate            datetime,
Type                    uniqueidentifier,
SubType                 uniqueidentifier,
Id                      uniqueidentifier,
SiteId                  uniqueidentifier,
ParentId                uniqueidentifier,
ItemId                  int,
BatchId                 uniqueidentifier,
ItemGuid                uniqueidentifier,
WebId                   uniqueidentifier,
UserId                  int,
Created                 datetime,
BinaryPayload           image,
TextPayload             ntext,
InternalState           int;
```

DeliveryDate: A **UTC datetime** representing when a work item is scheduled for execution. MUST NOT be NULL.

Type: The **work item type identifier** of the **work item type**. MUST NOT be NULL.

SubType: The **work item subtype identifier** of the **work item subtype**.

Id: The **work item identifier**.

SiteId: The Site Collection identifier of the Site Collection.

ParentId: The **work item parent identifier** of the Work Item.

ItemId: An **item identifier** for an list item associated with the work item. SHOULD <1> be 0 if there is no associated item. MUST NOT be NULL.

BatchId: The **work item batch identifier** of the **work item batch**. MUST be NULL if and only if the Work Item is a **timer job**.

ItemGuid: The Item GUID.

WebId: The Site identifier (as specified in [MS-WSSFO], section 2.2.1.10) of the Site.

UserId: The User identifier (as specified in [MS-WSSFO], section 2.2.1.12) of the user associated with the work item. MUST NOT be NULL.

Created: The date and time in UTC specifying when the server created the Work Item.

BinaryPayload: A vendor defined binary that can be used by the work item during execution.

TextPayload: The work item **text payload**.

InternalState: An integer bit field specifying the internal state of the Work Item. All valid values are specified in the following table.

Value	Description
0x00000001	The Work Item is marked as in progress work item.
0x00000002	The Work Item is marked as completed work item.
0x00000004	The Work Item is marked for automatic deletion.
0x00000008	The Work Item is marked for **exponential backoff**.
0x00000010	The Work Item is marked for **throttled fetch**.

2.2.7 Tables and Views

None.

2.2.8 XML Structures

The syntax of the definitions in this section use XML Schema as specified in [XMLSCHEMA1] and [XMLSCHEMA2].

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

The following table summarizes the set of common **XML schema** element definitions in this specification.

2.2.8.4.1 Workflow Modifications

This is an XML structure that stores data about a workflow. The structure is used to store and correlate a set of vendor-supplied **GUIDs** and vendor-supplied **XML**.

```
<s:element name="Mods">
  <s:complexType>
    <s:sequence>
      <s:element name="Mod" minOccurs="0" maxOccurs="unbounded">
        <s:element name="SubId" type="s:string" minOccurs="1" maxOccurs="1"/>
        <s:element name="Id" type="s:string " minOccurs="1" maxOccurs="1"/>
        <s:element name="Data" type="s:string" minOccurs="1" maxOccurs="1"/>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
```

Mod.SubId: A string containing a GUID in which any alphabetic characters MUST be in upper case. This GUID is a vendor-extensible field.

Mod.Id: An upper-case alphabetic string containing a GUID identifying the parent **Mod** element.

Mod.Data: Any string or valid XML. This is a vendor–extensible field.

Example:

```
 <Mods>
  <Mod>
    <SubId>F9168C5E-CEB2-4FAA-B6BF-329BF39FA1E4</SubId>
    <Id>936DA01F-9ABD-4D9D-80C7-02AF85C822A8</Id>
    <Data>Data string<a/><b>c</b></Data>
  </Mod>
</Mods>
```

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

The back-end database server maintains the following sets of data for this protocol within both a **configuration database** and one or more **content databases**. Data within the appropriate databases is maintained until updated or removed.

3.1.1.1 Web Parts

3.1.1.1.1 Customizable and Personalizable Properties

A Web Part defines a number of properties that can be modified to change how the Web Part behaves or renders. The properties are split into two groups, customizable and personalizable. These two groups of properties are stored in the back end database server for each Web Part, and both sets of properties are used to instantiate and render a Web Part on a front-end Web server. It is up to the Web Part implementer to determine whether a property is customizable or personalizable. A property is customizable if all users accessing the Web Part MUST get the same value for the property. A property is personalizable if users accessing the Web Part MUST be able to modify the property to a value specific to each user.

3.1.1.1.2 Adding and Modifying a Web Part for All Users (Customization)

When a Web Part is added to the **shared view** of a Web Parts page a new entry for the Web Part is added into the back end database server containing all the personalizable and customizable properties of the Web Part. For each version of a Web Parts page there is only one copy of the personalizable and customizable properties stored in the back end database server for the shared view of a Web Part. As a result, when two different users browse to the shared view of the same Web Parts page the same set of personalizable and customizable properties for the Web Part are returned, resulting in the same Web Part being rendered for each user. Modifying this copy of properties used to render the shared view of a Web Part is called customization, and all users browsing to the shared view of the Web Parts page will see the same customized Web Part.

3.1.1.1.3 Adding a Web Part for All Users then modifying it uniquely for a particular User (Personalization)

When a Web Part is added to the shared view of a Web Parts page and a user then accesses the shared view or **personal view** of the Web Parts page, the personalizable and customizable properties returned for the Web Part will be the same so the Web Part will render the same in both the shared view and personal view.

If the user then modifies the Web Part from the personal view of the Web Parts page, then all of the personalizable properties currently stored in the back end database server for the Web Part are copied into a separate entry in the back end database server for the Web Part that is associated with the particular user who modified the Web Part.

This process is called personalization, and it means there are now two copies of the personalizable properties for the Web Part in the back end database server, one copy that is used when any user accesses the Web Part in the shared view of the Web Parts page or they access the Web Part in the personal view of the Web Parts page but have not yet personalized the Web Part, and a second copy that is used when the user who personalized the Web Part accesses the Web Part in the personal view of the Web Parts page.

Every time a different user personalizes the Web Part an additional copy of the personalizable properties are stored for the Web Part in the back end database server for that particular user. When a user accesses the personal view of a Web Parts page, personalizable and customizable properties for the Web Part will be returned. If the Web Part has not been personalized by this user then these properties will be the same ones that are returned if the user browsed to the shared view of the Web Parts page. If the Web Part has been personalized by this user then the personalizable properties will be a unique copy that is stored in the back end database server just for this user, the customizable properties will be the same ones that are returned when accessing the shared view of the Web Parts page. There is only one copy of the customizable properties of a Web Part for a particular version of a Web Parts page, there is one copy of the personalizable properties of a Web Part for each user who has personalized that Web Part on the Web Parts page.

3.1.1.1.4 Adding a Web Part just for a particular User (Personal Web Part)

When a Web Part is added to the personal view of a Web Parts page a new entry for the Web Part is added into the back end database server containing all the personalizable and customizable properties of the Web Part, and the entry is associated with the particular user who added the Web Part. This is called a **personal Web Part** and it will only be returned when the user who added the Web Part is accessing the Web Parts page in personal view. No one else will ever have access to this personal Web Part. If a personal Web Part is modified the one copy of the personalizable and customizable properties for the Web Part in the back end database server will be updated, and again only the user who added the personal Web Part will see the changes when they access the personal view of the Web Parts page.

3.1.1.1.5 Versioning Web Parts Pages

Versioning can be configured per list or per **document library** to store multiple versions of a Web Parts page. If **minor version control** is enabled on a Web Parts page, and modifications are made to a Web Part on that Web Parts page, then the back end database server creates and stores a new version of the Web Parts page. The changes will be attributed to the user who made the changes. When a new version of a Web Parts page is created, an additional copy of all the personalizable and customizable properties used to render the shared view of the Web Parts on the Web Parts page is also created in the back end database server, this allows the Web Parts for different versions of the same Web Parts page to be independently modified. If a Web Part has been personalized by a user an additional copy of that users personalizable properties is NOT created, when a new version of the Web Parts page is created, this means if there are multiple versions of a Web Parts page there is only one copy of a particular users personalizations that gets used when that user is accessing the personal view of different versions of the Web Parts page.

For more information, refer to [MS-WSSO], section 3.3.4, Versioning.

3.1.1.1.6 Changing a Web Part Type Identifier

If this protocol is used to change the Web Part type identifier of an existing Web Part then the metadata stored in the back end database server for that Web Part is no longer valid and is deleted.

3.1.1.1.7 Web Part Caching

Web Parts can choose to cache data to improve their performance or behavior on subsequent renderings. If this protocol is used to modify the properties of an existing Web Part that change potentially invalidates data that the Web Part has cached so if any such cached data exists for the Web Part it is deleted.

3.1.1.2 Workflow

3.1.1.2.1 Workflow Concepts

A workflow template defines a particular process of operations. The definition structures the order of operation, constraints, timing, and actual operations of this process. For example, a process which defines and manages how fields are changed on a **document** is a workflow template.

3.1.1.2.2 Workflow Reusability

A Workflow is based on a Workflow association that is applied to a specific list or Content type. Similarly a Workflow association is based on a workflow template, one of several processes stored on the server.

In line with this hierarchy, a workflow template creates one or many Workflow associations and a Workflow association creates one or many Workflows. This enables a particular process of operations to be reused in many different contexts.

3.1.1.3 Work Items

A Work item represents a unit of work that is scheduled for execution at the time indicated by its Work Item Delivery Date. Information about work items is kept in back end database server. The Work Item information specifies what type of work the work items will perform, when they MUST run, and what objects are related to them. These work items can be run by a protocol client that iterates through them and performs the appropriate code based on the work item type. Thus, a protocol client that creates the Work item works in tandem with a protocol client that retrieves and runs them in the way they were intended to be performed. Work item entries identified by work item identifiers.

3.1.1.4 Event Receivers

3.1.1.4.1 Event Receiver Concepts

Event receivers are custom code for extending functionalities by reacting to Events. Registration information about Event Receivers is kept in back end database server. The registration information determines what Event Receivers are processed for an Event.

3.1.1.4.2 Registering Event Receivers

The Event Host MUST register an Event Receiver to handle Events. If the event receiver is registered by feature or content type then the event receiver event receiver source property MUST point to this feature or content type, otherwise it MUST be NULL.

3.1.1.4.3 Scopes of Event Receivers

Event receivers can be registered on event hosts of different scopes. When an event is fired, it bubbles from the innermost event host outwards. For example, when a list item is updated, the

server fires an item updating event on the parent list containing the list Item first then on the Site containing the parent list.

3.1.1.4.4 Sequences of Event Receivers

When there are more than one event receiver registered on an event host, the processing order of the event receivers is the numerical order of the **sequence numbers (1)** of these event receivers. The event receiver with the smallest sequence number (1) is processed first.

3.1.1.5 Quota Management

Event, Web Part, Workflow, and Work Item operations typically use, or free, disk space in the back-end database server. To manage this limited resource, quota management features can be enabled to track disk space usage, and block Event, Web Part, Workflow, and Work Item operations that would use additional disk space if a Site Collection has exceeded its quota limits. See [MS-WSSCADM], section 1.3.2, for more information about quota management.

3.1.2 Timers

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

3.1.3 Initialization

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

3.1.4 Message Processing Events and Sequencing Rules

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

For 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_AddNonListViewFormPersonalization

The proc_AddNonListViewFormPersonalization stored procedure is called to add a personalization to an existing Web Part which is not a list view Web Part. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_AddNonListViewFormPersonalization(
    @SiteId                 uniqueidentifier,
    @DocId                  uniqueidentifier,
    @WebPartId              uniqueidentifier,
    @UserId                 int,
    @PartOrder              int,
    @ZoneId                 nvarchar(64),
```

```
    @IsIncluded              bit,
    @FrameState              tinyint,
    @UserProperties          image
);
```

@SiteId: The Site Collection identifier of the Site Collection which contains the Web Part to be personalized. This MUST NOT be NULL.

@DocId: The document identifier (as specified in [MS-WSSFO], section 2.2.1.2) of the Document which contains the Web Part to be personalized. This MUST NOT be NULL.

@WebPartId: The Web Part identifier (as specified in [MS-WSSFO], section 2.2.1.14) of the Web Part to be personalized. This MUST NOT be NULL.

@UserId: The User identifier (as specified in [MS-WSSFO], section 2.2.1.12) of the user which personalizes the Web Part. This MUST NOT be NULL.

@PartOrder: The Web Part Zone Index of the added Web Part.

@ZoneId: The Web Part Zone identifier of the **Web Part zone** in which to put the Web Part.

@IsIncluded: The *Web Part Is Closed* state of the added Web Part. This value MUST NOT be NULL.

@FrameState: The *Web Part Chrome* state of the added Web Part. This MUST NOT be NULL.

@UserProperties: The Web Part properties to assign to this Web Part for the user specified by @UserId.

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

Value	Description
0	Successful completion.
1	The operation failed to complete.
2	The requested Web Part does not exist.
212	The specified Site Collection is Locked.
1816	The Quota for the specified Site Collection has been exceeded.

Result Sets: MUST NOT return any result sets.

3.1.4.2 proc_AddNonListViewFormWebPartForUrl

The proc_AddNonListViewFormWebPartForUrl stored procedure is called to add a Web Part that is not a **List View Web Part** or **List Form Web Part** to a given **page**. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_AddNonListViewFormWebPartForUrl(
    @SiteId                         uniqueidentifier,
    @DocDirName                     nvarchar(256),
    @DocLeafName                    nvarchar(128),
    @WebPartId                      uniqueidentifier,
```

```
    @ListId                         uniqueidentifier,
    @Type                           tinyint,
    @Flags                          int,
    @DisplayName                    nvarchar(255),
    @ContentTypeId                  varbinary(512),
    @Version                        int,
    @PartOrder                      int,
    @ZoneId                         nvarchar(64),
    @IsIncluded                     bit,
    @FrameState                     tinyint,
    @WebPartTypeId                  uniqueidentifier,
    @AllUsersProperties             image,
    @PerUserProperties              image,
    @Cache                          image,
    @Source                         ntext,
    @UserId                         int = NULL,
    @Level                          tinyint = 1,
    @BaseViewId                     tinyint = NULL,
    @bHasFGP                        bit = NULL,
    @bDeleteUsersOtherWebParts      bit = 0,
    @View                           ntext = NULL
);
```

@SiteId: The Site Collection identifier of the Site Collection which contains the Web Part Page to which to add the Web Part. This MUST NOT be NULL.

@DocDirName: The **directory name** of the Web Part Page to which to add the Web Part. This MUST NOT be NULL.

@DocLeafName: The **leaf name** of the Web Part Page to which to add the Web Part. This MUST NOT be NULL.

@WebPartId: The Web Part identifier (as specified in [MS-WSSFO], section 2.2.1.14) of the Web Part being added. MUST NOT be NULL.

@ListId: The list identifier (as specified in [MS-WSSFO], section 2.2.1.5) of the list with which to associate the Web Part.

@Type: The Page type (as specified in [MS-WSSFO], section 2.2.3.12) for the **list view**. If @Type has a value of **default view** the **view** MUST be made the Default View for the list.

@Flags: The set of View Flags (as specified in [MS-WSSFO], section 2.2.2.11) to be applied to the added Web Part.

@DisplayName: The Display Name of the Web Part being added.

@ContentTypeId: The Content type identifier (as specified in [MS-WSSFO], section 2.2.1.1) of the list Items in the list to be displayed in the Web Part.

@Version: The version number of the Web Part to add.

@PartOrder: The Web Part Zone Index of the Web Part to add.

@ZoneId: The Web Part Zone identifier of the Web Part Zone of the Web Part being added.

@IsIncluded: The Web Part Is Closed State of the added Web Part.

@FrameState: The Web Part Chrome State of the added Web Part.

@WebPartTypeId: The Web Part type identifier of the Web Part being added. MUST NOT be NULL.

@AllUsersProperties: A serialized representation of zero or more customizable properties on the Web Part.

@PerUserProperties: A serialized representation of zero or more personalizable properties on the Web Part.

@Cache: Private data cache of the Web Part.

@Source: The Web Part properties of the Web Part in either WPV2:WebPart format (as specified in [MS-WPPS], section 2.2.4.2) or HTML format.

@UserId: The User identifier (as specified in [MS-WSSFO], section 2.2.1.12) for the **current user**.

@Level: The publishing level of the Web Part Page for the current user.

@BaseViewId: The **base view identifier** for this Web Part.

@bHasFGP: This parameter MUST be ignored by protocol server.

@bDeleteUsersOtherWebParts: This parameter specifies whether all Web Parts on the page registered to the user MUST be deleted before this Web Part is added. If set to "1" then all Web Parts on the Page defined by the @SiteId, @Level, @DocDirName, and @DocLeafName parameters and registered to the user identified by @UserId MUST be **deleted** before this Web Part is added. If set to "0" then other Web Parts MUST NOT be modified.

@View: The **CAML** XML for the View to be applied to the Web Part.

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

Value	Description
0	Successful completion.
1	An **SQL** error occurred.
2	The specified Web Part Page cannot be found or @SiteId, @DirName or @LeafName is NULL.
212	The specified Site Collection is Locked.
1816	The Quota for the specified Site Collection has been exceeded.

Result Sets: MUST NOT return any result sets.

3.1.4.3 proc_AddWebPart

The proc_AddWebPart stored procedure is called to add a Web Part to a Web Part Page. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_AddWebPart(
    @SiteId                 uniqueidentifier,
    @DirName                nvarchar(256),
    @LeafName               nvarchar(128),
    @Level                  tinyint OUTPUT,
    @bAllUser               bit,
    @UserId                 int,
```

```
    @WebPartID              uniqueidentifier,
    @WebPartTypeID          uniqueidentifier,
    @bCheckLock             bit,
    @IsIncluded             bit,
    @FrameState             tinyint,
    @ZoneID                 nvarchar(64),
    @PartOrder              int,
    @AllUsersProperties     image,
    @PerUserProperties      image
);
```

@SiteId: The Site Collection identifier of the Site Collection which contains the Web Part Page to which to add the Web Part. This value MUST NOT be NULL.

@DirName: The Directory Name of the Web Part Page to which to add the Web Part. This value MUST NOT be NULL.

@LeafName: The Leaf Name of the Web Part Page to which to add the Web Part. This value MUST NOT be NULL.

@Level: The publishing level of the Web Part Page. The value is returned as an output parameter and MUST be the same value passed in or **draft**. The value is changed to Draft if the Web Part Page is in a Document Library, @Level is **published**, @bCheckLock is 1, @bAllUser is 1, @UserId references an existing user in the Site Collection, the Web Part Page is **moderated** or has minor version control enabled, and creation of a new version of the Web Part Page succeeded.

@bAlluser: If this flag is set to 1 the Web Part is added to the Shared View of the Web Part Page and is available to **All Users**. If this flag is set to 0, @UserId is used to add the Web Part to the current user's personal View of the Web Part Page and is available only to the current user.

@ UserId: The User identifier (as specified in [MS-WSSFO], section 2.2.1.12) for the current user. If the Web Part Page is Moderated or has minor version control enabled then @UserId is used to track who is adding the Web Part.

@WebPartID: The Web Part identifier (as specified in [MS-WSSFO], section 2.2.1.14) of the Web Part being added. This value MUST NOT be NULL.

@WebPartTypeID: The Web Part type identifier of the Web Part being added. This value MUST NOT be NULL.

@bCheckLock: If this flag is set to 1, check if the document is in a state where it can be modified, if it cannot be modified, return specific return code values, defined in the following Return Code Values table, that explain why it cannot be modified. If this flag is set to 0, the checks made when this flag is set to 1, are bypassed.

@IsIncluded: The Web Part Is Closed State of the Web Part.

@FrameState: The Web Part Chrome State of the Web Part.

@ZoneID: The name of the Web Part Zone identifier of the Web Part Zone to which to add the Web Part.

@PartOrder: The Web Part Zone Index of the Web Part.

@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.

@PerUserProperties: A serialized representation of 0 or more personalizable properties on the Web Part. If this value is NULL then default values will be used for all of the personalizable properties on the Web Part.

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

Value	Description
0	Successful completion.
1	An error occurred executing the stored procedure
2	The Web Part Page cannot be found.
3	The Web Part Page is in a **Document Library**, @Level is Published, @bCheckLock is 1, @bAllUser is 1, @UserId references an existing user in the Site Collection, the Web Part Page is Moderated or has minor version control enabled, and a new Draft version of the Web Part Page cannot be created because a unique name for it cannot be created.
12	@bCheckLock is 1, @bAllUser is 0 and the Web Part Page is Checked Out.
33	@bCheckLock is 1, @bAllUser is 1, and the specified Web Part Page is not the **current version**.
87	The Web Part Page is in a **Document Library**, @Level is Published, @bCheckLock is 1, @bAllUser is 1, @UserId references an existing user in the Site Collection, the Web Part Page is Moderated or has minor version control enabled, and a new Draft version of the Web Part Page cannot be created.
158	@bCheckLock is 1, @bAllUser is 1, the Web Part Page is in a Document Library with **Required Checkout** set and it is not Checked Out.
160	The Web Part Page is in a **Document Library**, @Level is Published, @bCheckLock is 1, @bAllUser is 1, the Web Part Page is Moderated or has minor version control enabled, and @UserId is NULL.
212	The Site Collection is Locked.
1816	The **Quota** for the Site Collection has been exceeded.

Result Sets: MUST NOT return any result sets.

3.1.4.4 proc_AddWorkflow

The proc_AddWorkflow stored procedure is called to create a Workflow and add it to a list Item. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE dbo.proc_AddWorkflow (
    @WorkflowTemplateId       uniqueidentifier,
    @WorkflowInstanceId       uniqueidentifier,
    @SiteId                   uniqueidentifier,
    @WebId                    uniqueidentifier,
    @ListId                   uniqueidentifier,
    @ItemId                   int,
    @Level                    tinyint,
    @Version                  int,
    @TaskListId               uniqueidentifier,
    @AdminTaskListId          uniqueidentifier,
    @Author                   int,
    @ProcessingId             uniqueidentifier,
    @InstanceData             image,
```

```
    @InstanceDataSize          int,
    @Modifications             ntext,
    @StatusFieldOrdinal        int,
    @StatusField               nvarchar(64)
);
```

@WorkflowTemplateId: The Workflow Template identifier of the workflow template on which the Workflow being added is based.

@WorkflowInstanceId: The Workflow identifier of the Workflow being added, or NULL. There MUST NOT be an existing workflow with the same workflow identifier. If @WorkflowInstanceId is NULL, the server MUST create a new identifier for the workflow. The server MUST set the creation and modification dates and times of the Workflow to the date and time in UTC the stored procedure is called.

@SiteId: The Site Collection identifier of the Site Collection which contains the Workflow. This value MUST NOT be NULL.

@WebId: The Site identifier, as specified in [MS-WSSFO], section 2.2.1.10, of the Site which contains the Workflow. This value MUST NOT be NULL.

@ListId: The **list identifier** of the list which contains the Workflow. This value MUST NOT be NULL.

@ItemId: The list Item identifier (as specified in [MS-WSSFO], section 2.2.1.6) of the list Item the Workflow is created for. This value MUST NOT be NULL.

@Level: If @ListId represents a Document Library, this represents the **publishing level** of the Document corresponding to the list Item specified by @ItemId. If @List does not represent a Document Library, this MUST be 1.

@Version: The list item version of the list Item, or the value 0.

@TaskListId: The list identifier of the Workflow Task list of the workflow. This value MUST NOT be NULL.

@AdminTaskListId: This parameter MUST be NULL.

@Author: The User identifier of the user creating the Workflow. This value MUST NOT be NULL.

@ProcessingId: This parameter MUST be NULL.

@InstanceData: This parameter MUST be NULL.

@InstanceDataSize: This parameter MUST contain the value 0.

@Modifications: This parameter MUST contain an empty string.

@StatusFieldOrdinal: The ordinal of the Workflow Status field of the Workflow. @StatusFieldOrdinal MUST be NULL if and only if @StatusField is null.

@StatusField: The name of the Workflow Status field of the workflow. The server MUST update the field specified by @StatusField and @StatusFieldOrdinal of the list Item specified by @ItemId to the Workflow identifier of the Workflow.

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

Value	Description
0	Successful completion.
2	No workflow association was found based on the workflow template specified by @WorkflowTemplateId in the site collection specified by @SiteId.
82	The workflow could not be added.
87	At least one input parameter was invalid.
183	The list item specified by @ItemId already has a workflow that is not a completed workflow.

Result Sets: MUST NOT return any result sets.

3.1.4.5 proc_AddWorkflowAssociation

The proc_AddWorkflowAssociation stored procedure is called to add a workflow association. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE dbo.proc_AddWorkflowAssociation(
    @Id                     uniqueidentifier,
    @BaseId                 uniqueidentifier,
    @ParentId               uniqueidentifier,
    @Name                   nvarchar(255),
    @Description            nvarchar(1023),
    @StatusFieldName        nvarchar(64),
    @SiteId                 uniqueidentifier,
    @WebId                  varbinary(16),
    @ListId                 varbinary(16),
    @ContentTypeId          varbinary(512),
    @TaskListId             varbinary(16),
    @HistoryListId          varbinary(16),
    @TaskListTitle          nvarchar(255),
    @HistoryListTitle       nvarchar(255),
    @Author                 int,
    @Configuration          int,
    @AutoCleanupDays        int,
    @PermissionsManual      bigint,
    @InstantiationParams    ntext
);
```

@Id: The Workflow association identifier of the Workflow association being created. If this value is NULL, the server MUST create a new Workflow association identifier for the Workflow association. The server MUST set the creation and modification times for the workflow association to the date and time in UTC when the procedure was called.

@BaseId: The Workflow Template identifier of the workflow template on which the Workflow association is based. This value MUST NOT be NULL.

@ParentId: The Workflow association identifier of the parent Workflow association of the Workflow association.

@Name: The name of the Workflow association.

@Description: The description of the Workflow association.

@StatusFieldName: The name of the Workflow Status field of the Workflow association.

@SiteId: The Site Collection identifier of the Site Collection which contains the Workflow association. This value MUST NOT be NULL.

@WebId: The Site identifier (as specified in [MS-WSSFO], section 2.2.1.10) of the Site which contains the Workflow association. This value MUST NOT be NULL.

@ListId: The list identifier of the list with which the Workflow association is associated.

@ContentTypeId: The Content type identifier (as specified in [MS-WSSFO], section 2.2.1.1) of the Content type with which the Workflow is associated.

@TaskListId: The list identifier of the Workflow Task list of the Workflow association.

@HistoryListId: The list identifier of the Workflow History list of the Workflow association.

@TaskListTitle: The title of the Workflow Task list of the Workflow association.

@HistoryListTitle: The title of the Workflow History list of the Workflow association.

@Author: The User identifier of the Author of the workflow.

@Configuration: The Workflow association Configuration of the Workflow association.

@AutoCleanupDays: The number of days before Workflows based on the Workflow association are cleaned up. MUST contain a positive integer.

@PermissionsManual: The WSS Rights Mask required to manually start any Workflows created from the Workflow association.

@InstantiationParams: The workflow association data of the Workflow association.

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

Value	Description
0	Successful completion.
82	The workflow association was not created.

Result Sets: MUST NOT return any result sets.

3.1.4.6 proc_AddWorkItem

The proc_AddWorkItem stored procedure is called to add a new Work Item to the set of pending work items. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_AddWorkItem(
    @WorkItemId                     uniqueidentifier,
    @DeliveryDate                   datetime,
    @Type                           uniqueidentifier,
    @SubType                        uniqueidentifier,
    @SiteId                         uniqueidentifier,
    @ParentId                       varbinary(16),
    @ItemId                         int,
    @WebId                          varbinary(16),
    @ItemGuid                       varbinary(16),
```

```
    @BatchId                        varbinary(16),
    @UserId                         int,
    @BinaryPayload                  image,
    @TextPayload                    ntext,
    @ProcessingId                   uniqueidentifier,
    @AutoDeleteOld                  bit = 0,
    @ExponentialRetryBackOff        bit = 1
);
```

@WorkItemId: The Work Item identifier of the Work Item. If the parameter is not NULL, then the server MUST give the new Work Item a Work Item identifier equal to the value of the parameter. If the parameter is NULL, then the server MUST generate a GUID for the Work Item identifier.

@DeliveryDate: The Work Item Delivery Date. If the parameter is NULL, then the server MUST schedule the Work Item to run immediately.

@Type: The Work Item type identifier of the Work Item type. MUST NOT be NULL.

@SubType: The Work Item Subtype identifier of the Work Item Subtype or the **empty GUID**.

@SiteId: The Site Collection identifier of the Site Collection or the Empty GUID. MUST NOT be NULL.

@ParentId: The Work Item Parent identifier of the Work Item. MUST NOT be NULL.

@ItemId: An Item identifier for an list item associated with the work item. SHOULD <2> be 0 if there is no associated item. MUST NOT be NULL.

@WebId: The Site identifier (as specified in [MS-WSSFO], section 2.2.1.10) of the Site.

@ItemGuid: The Item GUID of the list Item or the Empty GUID.

@BatchId: The Work Item Batch identifier of the Work Item Batch of the Work Item or the Empty GUID.

@UserId: The User identifier (as specified in [MS-WSSFO], section 2.2.1.12) of the user associated with the work item.

@BinaryPayload: The work item **binary payload**.

@TextPayload: The work item text payload.

@ProcessingId: The identifier of the **work item process**. If this parameter is NULL, then the client MUST NOT process the Work Item.

@AutoDeleteOld: This parameter specifies whether calls to proc_RevertInProgressWorkItem or proc_RevertInProgressWorkItems MUST cause the server to delete this Work Item if it has a Work Item Delivery Date 10 or more days prior to the call. A value of 0 specifies that the deletion MUST NOT occur. A value of 1 specifies that the deletion MUST occur.

@ExponentialRetryBackOff: This parameter specifies whether or not the server MUST retry execution with Exponential Back-off from the Work Item Delivery Date when the client calls proc_RevertInProgressWorkItem or proc_RevertInProgressWorkItems. A value of 0 indicates that Exponential Back-off MUST NOT occur on retry. A value of 1 indicates that Exponential Back-off MUST occur on retry. A value of 1 indicates that Exponential Back-off MUST occur on retry.

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

Value	Description
0	Successful completion.
5	Error: Access denied.
82	Error: Failed to add the Work Item.

Result Sets: MUST NOT return any result sets.

3.1.4.7 proc_ApplyViewToListWebPart

The proc_ApplyViewToListWebPart stored procedure is called to apply the specified View to the specified list View Web Part. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ApplyViewToListWebPart(
    @SiteId                 uniqueidentifier,
    @DirName                nvarchar(256),
    @LeafName               nvarchar(128),
    @Level                  tinyint OUTPUT,
    @WebPartID              uniqueidentifier,
    @ViewId                 uniqueidentifier,
    @UserId                 int,
    @ViewEditPerms          int,
    @ViewBody               ntext,
    @ViewFlags              int OUTPUT
);
```

@SiteId: The Site Collection identifier of the Site Collection which contains the Web Part specified by @WebPartID. MUST NOT be NULL.

@DirName: The Directory Name of the Web Part Page that contains the Web Part specified by @WebPartID. MUST NOT be NULL.

@LeafName: The Leaf Name of the Web Part Page that contains the Web Part specified by @WebPartID. MUST NOT be NULL.

@Level: This is an input/output parameter. On input, this is the Publishing Level value of the Page specified by @LeafName that contains the Web Part specified by @WebPartID. On output, this is the Publishing Level of the Page specified by @LeafName after the specified View has been applied to the Web Part specified by @WebPartID. MUST NOT be NULL.

@WebPartID: The Web Part identifier (as specified in [MS-WSSFO], section 2.2.1.14) of the Web Part on which to apply the specified View. MUST NOT be NULL.

@ViewId: If @ViewId is not NULL, it is the GUID for a list View Web Part. The base view identifier, Content type identifier (as specified in [MS-WSSFO], section 2.2.1.1), and View Flags (as specified in [MS-WSSFO], section 2.2.2.11) from the Web Part specified by @ViewId MUST be copied to the Web Part specified by @WebPartID. If @ViewId is NULL, the base view identifier MUST be set to 0 on the Web Part specified by @WebPartID. The View Flags (as specified in [MS-WSSFO], section 2.2.2.11) MUST be copied from @ViewFlags to the Web Part specified by @WebPartID. The Content type identifier (as specified in [MS-WSSFO], section 2.2.1.1) of the Web Part specified by @WebPartID MUST NOT be changed. If @ViewId is NULL or not NULL, the VIEWFLAG_HIDDEN (0x00000008) View Flags (as specified in [MS-WSSFO], section 2.2.2.11) on the Web Part specified by @WebPartID MUST be set. The VIEWFLAG_PERSONAL (0x00040000) bit on the Web Part specified by @WebPartID MUST be set if the Web Part is a personal Web Part or cleared otherwise.

@UserId: The User identifier (as specified in [MS-WSSFO], section 2.2.1.12) for the user that is applying the specified View to the Web Part specified by @WebPartID. MUST NOT be NULL.

@ViewEditPerms: The set of **permission** flags for the User specified by @UserId. MUST NOT be NULL. @ViewEditPerms MUST be a bitwise logical combination of the values listed in the following table:

Value	Description
0x00000001	Add a personal View to a **list view page**
0x00000002	Add a personal View to a Web Part Page other than a list View Page
0x00000004	Add a **public view** to a list View Page
0x00000008	Add a Public View to a Web Part Page other than a list View Page
0x00000010	Modify a personal View on a list View Page
0x00000020	Modify a personal View on a Web Part Page other than a list View Page
0x00000040	Modify a Public View on a list View Page
0x00000080	Modify a Public View on a Web Part Page other than a list View Page

@ViewBody: The CAML for the View to be applied to the Web Part specified by @WebPartID.

@ViewFlags: This is an input/output parameter. On input, if @ViewId is NULL, this set of View Flags (as specified in [MS-WSSFO], section 2.2.2.11) MUST be applied to the Web Part specified by @WebPartID. The VIEWFLAG_HIDDEN (0x00000008) bit MUST be set. The VIEWFLAG_PERSONAL (0x00040000) bit MUST be set if the Web Part is a personal Web Part or cleared otherwise. On input, if @ViewId is not NULL, @ViewFlags MUST be ignored. On output, this is the set of View Flags (as specified in [MS-WSSFO], section 2.2.2.11) of the Web Part specified by @WebPartID after the specified View has been applied.

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

Value	Description
0	Successful completion.
1	Internal SQL error.
2	The Page specified by @DirName, @LeafName, and @Level does not exist or has been deleted; or the Web Part specified by @WebPartID does not exist or has been deleted.
3	The Web Part Page is Moderated or has minor version control enabled, and a new version of the Web Part Page cannot be created because a unique name for it cannot be created.
5	The user specified by @UserId does not have the necessary Permissions to modify the Web Part specified by @WebPartID.
12	Cannot modify a personal Web Part on a Page that is Checked Out.
33	The Page specified by @DirName, @LeafName, and @Level is not the Current Version.
87	The Page specified by @DirName, @LeafName, and @Level does not exist or has been deleted.

Value	Description
158	The Page specified by @DirName and @LeafName needs to be Checked Out because the Page lives in a Document Library with **Required Checkout** set.
160	Need to create a new version of the Page specified by @DirName and @LeafName, but no user is specified by @UserId.
212	Need to create a new version of the Page specified by @DirName and @LeafName, but the Site Collection specified by @SiteId is Locked.
1816	Need to create new version of Page specified by @DirName and @LeafName, but the Site Collection specified by @SiteId has exceeded its Quota.
-2147467259	An error occurred while the stored procedure was running.

Result Sets: MUST NOT return any result sets.

3.1.4.8 proc_AutoCleanupWorkflows

The proc_AutoCleanupWorkflows stored procedure is called to clean up completed workflows. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_AutoCleanupWorkflows();
```

Parameters: None.

Return Code Values: This stored procedure returns an integer Return Code which the client MUST ignore.

Result Sets: MUST NOT return any result sets.

3.1.4.9 proc_AutoDropWorkflows

The proc_AutoDropWorkflows stored procedure is called to delete workflows and workflow associations. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE dbo.proc_AutoDropWorkflows (
    @SiteId                 uniqueidentifier,
    @WebId                  uniqueidentifier,
    @ListId                 uniqueidentifier,
    @ListItemId             int,
    @TemplateId             uniqueidentifier,
    @AutoCleanupDate        datetime,
    @ForceDelete            int,
    @TopBeforeQuick         int = 2147483647
);
```

@SiteId: The Site Collection identifier of the Site Collection which contains the Workflows and Workflow associations. This value MUST NOT be NULL. The server MUST update the Site Collection Quota to remove the space used by the deleted Workflows.

@WebId: The Site identifier (as specified in [MS-WSSFO], section 2.2.1.10) of the Site which contains the Workflows and Workflow associations. If this value is NULL, the server MUST include all Sites.

@ListId: The list identifier of the list that is associated with the Workflows and Workflow associations. If this value is NULL, the server MUST include all lists.

@ListItemId: The list Item identifier (as specified in [MS-WSSFO], section 2.2.1.6) of the list for which the Workflows were created. If @ListId is NULL, this value MUST be NULL. If this value is NULL, the server MUST include all list Items.

@TemplateId: The Workflow Template identifier of the workflow template of the Workflow associations. If this value is NULL, the server MUST include all workflow templates.

@AutoCleanupDate: The date and time limit for deleting Workflow associations and Workflows. If @ForceDelete contains the value 1, the server MUST ignore @AutoCleanupDate. If @ForceDelete contains the value 0, @AutoCleanupDate MUST contain a valid date value that occurs before the date and time that proc_AutoDropWorkflows was called.

@ForceDelete: This parameter determines whether to delete all Workflows or only those that are complete and were last modified before @AutoCleanupDate. This value MUST be 0 or 1. When @ForceDelete is 0, the server MUST delete only Workflows that are complete and were last modified before the date specified by @AutoCleanupDate. When @ForceDelete is 1, the server MUST delete all workflows up to the @TopBeforeQuick limit meeting the criteria specified by @SiteId, @WebId, @ListId, @ListItemId and @TemplateId, and MUST ignore completion and modification date.

@TopBeforeQuick: This parameter limits the number of Workflows being deleted. This value MUST contain a positive integer or 0. The server MUST NOT delete more workflows than the number specified by @TopBeforeQuick. When @ForceDelete is 1, the @TopBeforeQuick limit is reached, and @TemplateId is not null, the server MUST mark all Workflow associations in the Site Collection specified by @SiteId and based on the workflow template specified by @TemplateId for deferred deletion by proc_AutoCleanupWorkflows. When @ForceDelete is 1, the @TopBeforeQuick limit is reached, and @TemplateId is null, the server MUST mark all workflows that were not deleted and that meet the criteria specified by the @SiteId, @WebId, @ListId and @ListItemId and @TemplateId parameters for deferred deletion by proc_AutoCleanupWorkflows.

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

Value	Description
0	Successful completion; @ForceDelete is 0, or @ForceDelete is 1 and some of the workflows meeting the criteria specified by the input parameters were not deleted because of the @TopBeforeQuick limit.
1	Successful completion; @ForceDelete is 1, and all workflows meeting the criteria specified by the input parameters were deleted.

Result Sets: MUST NOT return any result sets.

3.1.4.10 proc_CancelDeclarativeWorkflows

The proc_CancelDeclarativeWorkflows stored procedure is called to cancel all **declarative workflows** in a site collection. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_CancelDeclarativeWorkflows(
    @SiteId                  uniqueidentifier
```

```
);
```

@SiteId: The Site Collection identifier of the Site Collection which contains the Workflows.

Return Code Values: An integer which MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.4.11 proc_CancelWorkflow

The proc_CancelWorkflow stored procedure is called to cancel a workflow. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_CancelWorkflow(
    @SiteId                 uniqueidentifier,
    @WorkflowInstanceId     uniqueidentifier
);
```

@SiteId: The Site Collection identifier of the Site Collection which contains the Workflow.

@WorkflowInstanceId: The Workflow identifier of the Workflow being canceled. If the Workflow is finished or terminated, the stored procedure MUST NOT update the Workflow. The server MUST add the WFS_CANCELED flag to the workflow internal state and MUST remove the WFS_RUNNING, WFS_LOCKED and WFS_HASNEWEVENTS flags. If the Workflow Status1 field of the Workflow is WFSTAT_FAILEDTOSTART_RETRY, the server MUST set the Workflow Status1 field to WFSTAT_FAILEDTOSTART; otherwise, the server MUST set the Workflow Status1 field to WFSTAT_CANCELED. The server MUST delete any list Items in the Workflow Task list for the Workflow and any work items scheduled to process the Workflow. The server MUST set the modification date and time of the workflow to the date and time in UTC when the procedure was called.

Return Code Values: An integer which MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.4.12 proc_CompleteInProgressWorkItems

The proc_CompleteInProgressWorkItems stored procedure is called to mark a set of Work Items as Completed Work Items. The server MUST restrict the set to those specified by the parameters and for which the Work Item Delivery Date has passed. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_CompleteInProgressWorkItems(
    @ProcessingId           uniqueidentifier,
    @SiteId                 uniqueidentifier,
    @ParentId               uniqueidentifier,
    @WorkItemType           uniqueidentifier,
    @BatchId                uniqueidentifier
);
```

@ProcessingId: The Work Item Process identifier of the Work Item Process. The server MUST only modify Work Items associated with this Work Item Process.

@SiteId: The Site Collection identifier of the Site Collection. If the parameter is not NULL, then the server MUST only modify Work Items associated with this Site Collection. If the parameter is NULL, then the server MUST modify Work Items that meet the criteria specified by the other parameters regardless of associated Site Collection identifier.

@ParentId: The Work Item Parent identifier of the Work Item. If the parameter is not NULL, then the server MUST only modify Work Items which have this Work Item Parent identifier. If the parameter is NULL, then the server MUST modify Work Items that meet the criteria specified by the other parameters regardless of the value of their Work Item Parent identifier.

@WorkItemType: The Work Item type identifier of the Work Item type. The server MUST only modify Work Items associated with this Work Item type. MUST NOT be NULL.

@BatchId: The Work Item Batch identifier of the Work Item Batch. If the parameter is not NULL, then the server MUST only modify Work Items associated with this Work Item Batch. If the parameter is not NULL, then the server MUST modify Work Items that meet the criteria specified by the other parameters regardless of associated Work Item Batch identifier.

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

Result Sets: MUST NOT return any result sets.

3.1.4.13 proc_CopyDefaultViewWebParts

The proc_CopyDefaultViewWebParts stored procedure is called to copy Web Parts from the Shared View of a Web Part Page to a new Web Part Page. personal View Web Parts and the **default list view** Web Part are skipped. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_CopyDefaultViewWebParts(
    @SiteID                 uniqueidentifier,
    @DefaultViewDirName     nvarchar(256),
    @DefaultViewLeafName    nvarchar(128),
    @NewViewDocId           uniqueidentifier
);
```

@ SiteID: The Site Collection identifier of the Site Collection which contains the source and destination Web Part Pages. This parameter MUST NOT be NULL.

@ DefaultViewDirName: The Directory Name of the source Web Part Page. This parameter MUST NOT be NULL.

@ DefaultViewLeafName: The Leaf Name of the source Web Part Page. This parameter MUST NOT be NULL.

@ NewViewDocId: The GUID of the Web Part Page where the copied Web Parts will be placed. This parameter MUST NOT be NULL.

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

Value	Description
0	Successful completion.
2	The system cannot find the Web Part Page specified.
212	The specified Site Collection is Locked.

Value	Description
1150	A concurrency violation occurred. No such version of the Web Part Page exists.
1816	The **Quota** for the specified Site Collection has been exceeded.
-2147467259	An error occurred while the stored procedure was running.

Result Sets: MUST NOT return any result sets.

3.1.4.14 proc_CountWorkflowAssociations

The proc_CountWorkflowAssociations stored procedure is called to obtain a count of workflow associations for one or all workflow templates contained in a site collection. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_CountWorkflowAssociations(
    @SiteId            uniqueidentifier,
    @BaseId            uniqueidentifier
);
```

@SiteId: The Site Collection identifier of the Site Collection which contains the Workflow associations.

@BaseId: The Workflow Template identifier of the workflow template the Workflow associations are based on. If this value is NULL, the server MUST include all workflow templates.

Return Code Values: An integer which MUST be 0.

Result Sets: MUST return the following result set:

3.1.4.14.1 Count Workflow Associations Result Set

This Result Set contains exactly one row. The T-SQL syntax for the result set is as follows:

```
{Count}                int;
```

{Count}: The count of the workflow associations meeting the criteria specified by the input parameters.

3.1.4.15 proc_CountWorkflows

The proc_CountWorkflows stored procedure is called to obtain a count of workflows based on a workflow association or workflow template. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_CountWorkflows(
    @AssociationId            uniqueidentifier,
    @SiteId                   uniqueidentifier,
    @BaseId                   uniqueidentifier,
    @InternalState            int
);
```

@AssociationId: The Workflow association identifier of the Workflow association of the Workflows. If this value is not NULL, the server MUST ignore @SiteId and @BaseId. If this value is NULL, the server MUST include all Workflow associations.

@SiteId: The Site Collection identifier of the Site Collection which contains the Workflows.

@BaseId: The Workflow Template identifier of the workflow template the Workflows are based on.

@InternalState: A workflow internal state bitmask specifying the internal states of the Workflows. If @InternalState is not null, the server MUST restrict the count in the result set to Workflows which have at least one internal state flag in common with the bitmask, similar to Workflow.InternalState & @InternalState <> 0.

Return Code Values: An integer which MUST be 0.

Result Sets: MUST return the following result set:

3.1.4.15.1 Count Workflows Result Set

This Result Set contains exactly one row. The T-SQL syntax for the result set is as follows:

```
{Count}                int;
```

{Count}: The count of the workflows meeting the criteria specified by the input parameters.

3.1.4.16 proc_CountWorkflowsBatch

The proc_CountWorkflowsBatch stored procedure is called to obtain a set of workflow templates and the count of workflows based on each workflow template. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE dbo.proc_CountWorkflowsBatch(
    @SiteId                    uniqueidentifier,
    @WebId                     uniqueidentifier,
    @ListId                    uniqueidentifier,
    @ContentTypeId             uniqueidentifier,
    @InternalState             int
);
```

@SiteId: The Site Collection identifier of the Site Collection which contains the Workflows.

@WebId: The Site identifier (as specified in [MS-WSSFO], section 2.2.1.10) of the Site which contains the Workflows.

@ListId: The list identifier of the list which contains the Workflows. If @ListId is NULL, the server must include all lists.

@ContentTypeId: The Content type identifier (as specified in [MS-WSSFO], section 2.2.1.1) of the Content type from which the Workflows were created. If @ListId is not NULL, the server MUST ignore @ContentTypeId.

@InternalState: A workflow internal state bitmask specifying the internal states of the Workflows. If @InternalState is not NULL, the server MUST restrict the result to workflows which have an internal state that has at least one internal state flag in common with the bitmask (that is, Workflow.InternalState & @InternalState <> 0).

Return Code Values: An integer which MUST be 0.

Result Sets: MUST return the following result set:

3.1.4.16.1 Workflows Batch Result Set

This Result Set returns a set of workflow template identifiers and the count of workflows based on each template. The T-SQL syntax for the result set is as follows:

```
TemplateId              uniqueidentifier
{Count}                 int;
```

TemplateId: The Workflow Template identifier. At least one Workflow specified by the input parameters MUST be based on the workflow template.

{Count}: The count of the Workflows based on the workflow template. This value MUST be greater than zero.

3.1.4.17 proc_CreateListViewPart

The proc_CreateListViewPart **stored procedure** is called to add a list View Web Part to a Web Part Page. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_CreateListViewPart(
    @SiteId                 uniqueidentifier,
    @bAllUser               bit,
    @UserId                 int,
    @DirName                nvarchar(256),
    @LeafName               nvarchar(128),
    @Level                  tinyint OUTPUT,
    @WebPartID              uniqueidentifier,
    @WebPartTypeID          uniqueidentifier,
    @IsIncluded             bit,
    @FrameState             tinyint,
    @ZoneID                 nvarchar(64),
    @PartOrder              int,
    @ListId                 uniqueidentifier,
    @BaseViewId             uniqueidentifier,
    @Flags                  int,
    @ContentTypeId          tContentTypeId,
    @AllUsersProperties     image,
    @PerUserProperties      image,
    @View                   ntext,
    @DisplayName            nvarchar(255)
);
```

@SiteId: The Site Collection identifier of the Site Collection which contains the Web Part Page to which the list View Web Part will be added. MUST NOT be NULL.

@bAllUser: Specifies whether to add the Web Part for the Shared View or personal View of the Web Part Page. If this flag is set to 1 the Web Part is added to the Shared View of the Web Part Page and is available to All Users. If this flag is set to 0 @UserId is used to add the Web Part to the current user's personal View of the Web Part Page and is available only to the current user.

@UserId: The User identifier (as specified in [MS-WSSFO], section 2.2.1.12) for the current user. If the Web Part Page is moderated or has **version control** enabled then @UserId is used to track who is adding the Web Part.

@DirName: The **Directory Name** of the Web Part Page to which to add the list View Web Part. MUST NOT be NULL.

@LeafName: The **Leaf Name** of the Web Part Page to which to add the list View Web Part. MUST NOT be NULL.

@Level: The Publishing Level of the Web Part Page for the current user. The value is returned as an output parameter and MUST be the same value passed in or Draft. The value is changed to Draft if the Web Part Page is in a **Document Library**, @Level is Published, @bCheckLock is 1, @bAllUser is 1, @UserId references an existing user in the Site Collection, the Web Part Page is Moderated or has Version Control enabled, and creation of a new **version** of the Web Part Page succeeded.

@WebPartID: The Web Part identifier (as specified in [MS-WSSFO], section 2.2.1.14) of the Web Part being added. MUST NOT be NULL.

@WebPartTypeID: The Web Part type identifier of the Web Part being added. MUST NOT be NULL.

@IsIncluded: The Web Part Is Closed State of the added Web Part.

@FrameState: The **Web Part Chrome State** of the added Web Part.

@ZoneID: The Web Part Zone identifier of the Web Part Zone to which to add the Web Part.

@PartOrder: The Web Part Zone Index of the added Web Part.

@ListId: The List identifier (as specified in [MS-WSSFO], section 2.2.1.5) of the list for the Web Part

@BaseViewId: The base view identifier for the Web Part.

@Flags: A View Flags (as specified in [MS-WSSFO], section 2.2.2.11) value specifying **View** related settings for the Web Part.

@ContentTypeId: The Content type identifier (as specified in [MS-WSSFO], section 2.2.1.1) of the list Items in the list to be displayed in the Web Part.

@AllUsersProperties: A serialized representation of zero or more Customizable properties on the Web Part.

@PerUserProperties: A serialized representation of zero or more personalizable properties on the Web Part.

@View: CAML XML specifying View related settings for the Web Part.

@DisplayName: The Display Name for the Web Part.

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

Value	Description
0	Successful completion.
1	list View Web Part was not successfully created.

Value	Description
2	The specified Web Part Page cannot be found.
3	The Web Part Page is Moderated or has minor version control enabled, and a new version of the Web Part Page cannot be created.
12	Attempted to add a personalized list View Web Part to a Web Part Page whose Publishing Level is Checked Out.
33	The specified Web Part Page is not the Current Version.
87	The Web Part Page is in a **Document Library**, @Level is Published, , @bAllUser is 1, @UserId references an existing user in the Site Collection, the Web Part Page is Moderated or has minor version control enabled, and a new Draft version of the Web Part Page cannot be created
158	The Web Part Page is required to be Checked Out before it is modified and it is not Checked Out.
160	The Web Part Page is in a **Document Library**, @Level is Published, , @bAllUser is 1, the Web Part Page is Moderated or has minor version control enabled, and @UserId is NULL.
212	The specified Site Collection has been Locked.
1816	The **Quota** for the specified Site Collection has been exceeded.

Result Sets: MUST NOT return any result sets.

3.1.4.18 proc_DeleteDocEventReceiver

The **proc_DeleteDocEventReceiver** stored procedure is called to delete the registration of an event receiver for a specified document. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_DeleteDocEventReceiver(
    @DocUrl                 nvarchar(260),
    @Id                     uniqueidentifier,
    @Name                   nvarchar(256),
    @SiteId                 uniqueidentifier,
    @WebId                  uniqueidentifier,
    @ItemId                 int,
    @Type                   int,
    @SequenceNumber         int,
    @Assembly               nvarchar(256),
    @Class                  nvarchar(256),
    @Data                   nvarchar(256),
    @Filter                 nvarchar(256),
    @Credential             int
);
```

@DocUrl: The URL in store-relative form of the specified document that has the event receiver.

@Id: The event receiver identification of the event receiver.

@Name: The name of the event receiver.

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

@WebId: The Site identifier (as specified in [MS-WSSFO], section 2.2.1.10) of the site which contains the document.

@ItemId: Reserved. @ItemId MUST be 0.

@Type: The type of the event receiver. @Type MUST be a value of Event Receiver type (as specified in [MS-WSSFO], section 2.2.3.6).

@SequenceNumber: The sequence number (1) of the event receiver. @SequenceNumber MUST be greater than OR equal to zero 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: Reserved. @Filter MUST be NULL.

@Credential: Reserved. @Credential MUST be 0.

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

Value	Description
0	The event receiver identified by @Id is deleted from the site collection identified by @SiteId.
3	The document identified by @DocUrl is not found in the site identified by @WebId in the site collection identified by @SiteId.
87	The deletion failed.

Result Sets: MUST NOT return any result sets.

3.1.4.19 proc_DeleteEventReceiver

The proc_DeleteEventReceiver stored procedure is called to delete the registration of a specified event receiver. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_DeleteEventReceiver(
    @Id                     uniqueidentifier,
    @Name                   nvarchar(256),
    @SiteId                 uniqueidentifier,
    @WebId                  uniqueidentifier,
    @HostId                 uniqueidentifier,
    @HostType               int,
    @ItemId                 int,
    @DirName                nvarchar(256),
    @LeafName               nvarchar(128),
    @Type                   int,
    @SequenceNumber         int,
    @Assembly               nvarchar(256),
    @Class                  nvarchar(256),
    @Data                   nvarchar(256),
    @Filter                 nvarchar(256),
    @SourceId               varbinary(512),
    @SourceType             int,
    @Credential             int,
    @ContextType            uniqueidentifier,
    @ContextEventType       uniqueidentifier,
```

```
    @ContextId                uniqueidentifier,
    @ContextObjectId          uniqueidentifier,
    @ContextCollectionId      uniqueidentifier
);
```

@Id: The Event Receiver identifier (as specified in [MS-WSSFO], section 2.2.1.3) of the event receiver.

@Name: The name of the event receiver.

@SiteId: The Site Collection identifier of the Site Collection which contains the event host.

@WebId: The Site identifier (as specified in [MS-WSSFO], section 2.2.1.10) of the Site which contains the event host.

@HostId: The event host identifier of the event host of the event receiver.

@HostType: The type of the event host of the Event receiver. The value MUST be one of Event Host type (as specified in [MS-WSSFO], section 2.2.3.5).

@ItemId: Reserved. @ItemId MUST be 0.

@DirName: Reserved. @DirName MUST be NULL.

@LeafName: Reserved. @LeafName MUST be NULL.

@Type: The type of the event receiver. @Type MUST be one of Event Receiver type (as specified in [MS-WSSFO], section 2.2.3.6).

@SequenceNumber: The sequence number (1) 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: Reserved. @Filter MUST be NULL.

@SourceId: The event receiver source identifier of the event receiver. This is the Feature identifier (as specified in [MS-WSSFO], section 2.2.1.4) of the feature if the event receiver is added via a feature. This is the Content type identifier (as specified in [MS-WSSFO], section 2.2.1.1) of the content type if the event receiver is added via a content type. Otherwise, the event receiver source identifier MUST be NULL.

@SourceType: The Event Receiver Source type of the event receiver. @SourceType MUST be one of the Event Receiver Source Type values.

@Credential: Reserved. @Credential MUST be 0.

@ContextType: The context type identifier of the event receiver.

@ContextEventType: Reserved. @ContextEventType MUST be NULL.

@ContextId: The context identifier of the event receiver.

@ContextObjectId: The context object identifier for the Event Host of the event receiver.

@ContextCollectionId: The context collection identifier of the event receiver.

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

Value	Description
0	The event receiver whose identifier is @Id and type is not ContextLookupReceivers Event Receiver type (as specified in [MS-WSSFO], section 2.2.3.6) is successfully deleted from the site collection identified by @SiteId. All workflow event receivers associated with the workflow context identified by @ContextObjectId that are not used for active workflow are also deleted.
87	Delete failed because an event receiver whose identifier is @Id is not found OR whose type is ContextLookupReceivers Event Receiver type (as specified in [MS-WSSFO], section 2.2.3.6) in the site collection identified by @SiteId.

Result Sets: MUST NOT return any result sets.

3.1.4.20 proc_DeleteEventReceiversBySourceId

The proc_DeleteEventReceiversBySourceId stored procedure is called to delete the event receivers registered for a specified event host via a feature or content type. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_DeleteEventReceiversBySourceId(
    @SourceId                varbinary(512),
    @SourceType              int,
    @SiteId                  uniqueidentifier,
    @WebId                   uniqueidentifier,
    @HostId                  uniqueidentifier
);
```

@SourceId: The event receiver source identifier of the event receiver. This is the Feature identifier (as specified in [MS-WSSFO], section 2.2.1.4) of the feature if the event receiver is added via a feature. This is the Content type identifier (as specified in [MS-WSSFO], section 2.2.1.1) of the content type if the event receiver is added via a content type. Otherwise, the event receiver source identifier MUST be NULL.

@SourceType: The Event Receiver Source type of the event receivers to delete. @SourceType MUST be one of the Event Receiver Source Type values.

@SiteId: The Site Collection identifier of the site collection which contains the event host.

@WebId: The Site identifier (as specified in [MS-WSSFO], section 2.2.1.10) of the site which contains the event host.

@HostId: The event host identifier of the event host which the event receivers are associated with.

Return Code Values: An integer which MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.4.21 proc_DeleteInProgressWorkItems

The proc_DeleteInProgressWorkItems stored procedure is called to delete a set of Work Items that meet the criteria specified by the input parameter values. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_DeleteInProgressWorkItems(
    @ProcessingId            uniqueidentifier,
    @SiteId                  uniqueidentifier,
    @ParentId                uniqueidentifier,
    @WorkItemType            uniqueidentifier,
    @BatchId                 uniqueidentifier
);
```

@ProcessingId: The Work Item Process identifier of the Work Item Process. If the parameter is not NULL, then the server MUST only delete Work Items associated with this Work Item Process and for which the Work Item Delivery Date has passed. If the parameter is NULL, then the server MUST delete Work Items that meet the criteria specified by the other parameters regardless of associated Work Item Process or Work Item Delivery Date.

@SiteId: The Site Collection identifier of the Site Collection. The server MUST only delete Work Items associated with this Site Collection. MUST NOT be NULL.

@ParentId: The Work Item Parent identifier of the Work Item. If the parameter is not NULL, then the server MUST only delete Work Items which have this Work Item Parent identifier. If the parameter is NULL, then the server MUST delete Work Items that meet the criteria specified by the other parameters regardless of the value of their Work Item Parent identifer.

@WorkItemType: The Work Item type identifier of the Work Item type. The server MUST only delete Work Items associated with this Work Item type. MUST NOT be NULL.

@BatchId: The Work Item Batch identifier of the Work Item Batch. If the parameter is not NULL, then the server MUST only delete Work Items associated with this Work Item Batch. If the parameter is NULL, then the server MUST delete Work Items that meet the criteria specified by the other parameters regardless of associated Work Item Batch.

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

Result Sets: MUST NOT return any result sets.

3.1.4.22 proc_DeleteSmartPagePersonalization

The proc_DeleteSmartPagePersonalization stored procedure is called to delete personalizations from all Web Parts on the Web Part Page and to delete all personal Web Parts from the Web Part Page. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_DeleteSmartPagePersonalization(
    @SiteId                  uniqueidentifier,
    @DirName                 nvarchar(256),
    @LeafName                nvarchar(128),
    @UserId                  int
);
```

@SiteId: The Site Collection identifier for the Site Collection which contains the Page specified by @DirName and @LeafName. MUST NOT be NULL.

@DirName: The Directory Name of the Web Part Page from which to delete personalizations and personal Web Parts. MUST NOT be NULL.

@LeafName: The Leaf Name of the Web Part Page from which to delete personalizations and personal Web Parts. MUST NOT be NULL.

@UserId: The User identifier (as specified in [MS-WSSFO], section 2.2.1.12) for which to delete personalizations and personal Web Parts. If @UserId is NULL, the stored procedure MUST delete personalizations and personal Web Parts for every user. If @UserId is not NULL, the stored procedure MUST ONLY delete personalizations and personal Web Parts for the user specified by @UserId.

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

Value	Description
0	Successful completion.
1	Internal SQL error.
2	The Page specified by @DirName and @LeafName does not exist or has been deleted.

Result Sets: MUST NOT return any result sets.

3.1.4.23 proc_DeleteWebPart

The proc_DeleteWebPart stored procedure is called to delete Web Part from the Web Part Page. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_DeleteWebPart (
    @SiteId                 uniqueidentifier,
    @DirName                nvarchar(256),
    @LeafName               nvarchar(128),
    @Level                  tinyint OUTPUT,
    @UserId                 int,
    @WebPartID              uniqueidentifier
);
```

@SiteId: The Site Collection identifier for the Site Collection which contains the Web Part specified by @WebPartID. MUST NOT be NULL.

@DirName: The Directory Name of the Web Part Page that contains the Web Part specified by @WebPartID. MUST NOT be NULL.

@LeafName: The Leaf Name of the Web Part Page that contains the Web Part specified by @WebPartID. MUST NOT be NULL.

@Level: This is an input/output parameter. On input, this is the Publishing Level value of the Page specified by @LeafName that contains the Web Part specified by @WebPartID. On output, this is the Publishing Level of the Page specified by @LeafName after the Web Part specified by @WebPartID is deleted. MUST NOT be NULL.

@UserId: The User identifier (as specified in [MS-WSSFO], section 2.2.1.12) of the user which is deleting the Web Part specified by @WebPartID.

@WebPartID: The Web Part identifier (as specified in [MS-WSSFO], section 2.2.1.14) of the Web Part to be deleted. MUST NOT be NULL.

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

Value	Description
0	Successful completion.
1	Internal SQL error.
2	The Page specified by @DirName, @LeafName, and @Level does not exist or has been deleted.
3	The Web Part Page is Moderated or has minor version control enabled, and a new version of the Web Part Page cannot be created because a unique name for it cannot be created.
5	The Web Part specified by @WebPartID lives in a Page different from the Page specified by @DirName and @LeafName.
12	Cannot delete a personalized Web Part from a Page that is Checked Out.
33	The Page specified by @DirName, @LeafName, and @Level is not the Current Version.
87	The Page specified by @DirName, @LeafName, and @Level does not exist or has been deleted.
158	The Page specified by @DirName and @LeafName needs to be Checked Out because the Page lives in a Document Library with **Required Checkout** set.
160	Need to create a new version of the Page specified by @DirName and @LeafName, but no user is specified by @UserId.
212	Need to create a new version of the Page specified by @DirName and @LeafName, but the Site Collection specified by @SiteId is locked.
1816	Need to create new version of Page specified by @DirName and @LeafName, but the Site Collection specified by @SiteId has exceeded its quota.

Result Sets: MUST NOT return any result sets.

3.1.4.24 proc_DeleteWebPartPersonalization

The proc_DeleteWebPartPersonalization stored procedure is called to delete personalizations from the specified Web Part. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_DeleteWebPartPersonalization(
    @SiteId                 uniqueidentifier,
    @DirName                nvarchar(256),
    @LeafName               nvarchar(128),
    @UserId                 int,
    @WebPartID              uniqueidentifier
);
```

@SiteId: The Site Collection identifier for the Site Collection which contains the Web Part specified by @WebPartID. MUST NOT be NULL.

@DirName: The Directory Name of the Web Part Page that contains the Web Part specified by @WebPartID. MUST NOT be NULL.

@LeafName: The Leaf Name of the Web Part Page that contains the Web Part specified by @WebPartID. MUST NOT be NULL.

@UserId: The User identifier (as specified in [MS-WSSFO], section 2.2.1.12) of the user for which to delete personalizations. If @UserId is NULL, the stored procedure MUST delete personalizations from the Web Part specified in @WebPartID for every user. If @UserId is not NULL, the stored procedure MUST delete personalizations from the Web Part specified in @WebPartID for the user specified by @UserId.

@WebPartID: The Web Part identifier (as specified in [MS-WSSFO], section 2.2.1.14) of the Web Part from which to delete personalizations. MUST NOT be NULL.

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

Value	Description
0	Successful completion.
1	Internal SQL error.
2	The Page specified by @DirName and @LeafName does not exist or has been deleted.
5	The Web Part specified by @WebPartID lives in a Page different from the Page specified by @DirName and @LeafName.

Result Sets: MUST NOT return any result sets.

3.1.4.25 proc_DeleteWebPartWhileSaving

The proc_DeleteWebPartWhileSaving stored procedure is called to delete a Web Part from the Shared View of the Web Part Page. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_DeleteWebPartWhileSaving(
    @SiteId                 uniqueidentifier,
    @DirName                nvarchar(256),
    @LeafName               nvarchar(128),
    @Level                  tinyint,
    @WebPartID              uniqueidentifier
);
```

@SiteId: The Site Collection identifier of the Site Collection which contains the Web Part Page.

@DirName: The Directory Name of the Web Part Page.

@LeafName: The Leaf Name of the Web Part Page.

@Level: The Publishing Level of the Web Part Page.

@WebPartID: The Web Part identifier (as specified in [MS-WSSFO], section 2.2.1.14) of the Web Part to be deleted from the Web Part Page.

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

Value	Description
0	Successful completion.

Value	Description
2	The Web Part Page cannot be found or @SiteId, @DirName or @LeafName is NULL.
33	The Web Part Page is not the Current Version.
5	The Web Part is not in a Shared View.
1	An internal SQL error occurred.

Result Sets: MUST NOT return any result sets.

3.1.4.26 proc_DeleteZoneWebPartsWhileSaving

The proc_DeleteZoneWebPartsWhileSaving stored procedure is called to delete all the Web Parts in a Web Part Zone from the Shared View of the Web Part Page. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_DeleteZoneWebPartsWhileSaving(
    @SiteId                 uniqueidentifier,
    @DirName                nvarchar(256),
    @LeafName               nvarchar(128),
    @PageUrlID              uniqueidentifier,
    @Level                  tinyint,
    @WebPartZoneID          nvarchar(64)
);
```

@SiteId: The Site Collection identifier of the Site Collection which contains the Web Part Page.

@DirName: The Directory Name of the Web Part Page.

@LeafName: The Leaf Name of the Web Part Page.

@PageUrlID: The Document identifier (as specified in [MS-WSSFO], section 2.2.1.2) of the Web Part Page.

@Level: The Publishing Level of the Web Part Page.

@WebPartZoneID: The Web Part Zone identifier of the Web Part Zone of the Web Part Page.

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

Value	Description
0	Successful completion.
33	The Web Part Page is not the Current Version.
1	An internal SQL error occurred.

Result Sets: MUST NOT return any result sets.

3.1.4.27 proc_DisableAssociationsForTemplate

The proc_DisableAssociationsForTemplate stored procedure is called to disable Workflow associations based on a workflow template. When a Workflow association is disabled, no new

Workflows can be created from that Workflow association. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_DisableAssociationsForTemplate (
    @SiteId                 uniqueidentifier,
    @BaseId                 uniqueidentifier
);
```

@SiteId: The Site Collection identifier of the Site Collection.

@BaseId: The Workflow Template identifier of the workflow template. The server MUST disable all Workflow associations in the Site Collection based on the workflow template. The server MUST NOT allow any Workflow associations in the Site Collection based on the workflow template to create any new Workflows.

Return Code Values: An integer which MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.4.28 proc_DropWorkflow

The proc_DropWorkflow stored procedure is called to delete a workflow. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE dbo.proc_DropWorkflow(
    @WorkflowInstanceId     uniqueidentifier,
    @SiteId                 uniqueidentifier,
    @WebId                  uniqueidentifier,
    @ListId                 uniqueidentifier
);
```

@WorkflowInstanceId: The Workflow identifier of the Workflow. The server MUST delete the Workflow.

@SiteId: The Site Collection identifier of the Site Collection which contains the Workflow. The server MUST update the site collection quota to remove the space used by the deleted workflow.

@WebId: The protocol server MUST ignore this parameter.

@ListId: The protocol server MUST ignore this parameter.

Return Code Values: An integer which MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.4.29 proc_DropWorkflowAssociation

The proc_DropWorkflowAssociation stored procedure is called to delete a workflow association and its associated workflows. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE dbo.proc_DropWorkflowAssociation(
    @SiteId                 uniqueidentifier,
    @Id                     uniqueidentifier,
    @DropAll                int = 0
```

```
);
```

@SiteId: The Site Collection identifier of the Site Collection which contains the Workflow association. The server MUST update the site collection quota to remove the spaced used by the deleted workflows.

@Id: The Workflow association identifier of the Workflow association. The server MUST delete the Workflow association and all Workflows based on the Workflow association.

@DropAll: This parameter specifies whether the server throttles the deletion process. This value MUST be 0 or 1. When set to 1, the server MUST perform the entire deletion process immediately. When set to 0, the server MUST throttle the deletion process by deleting a limited number of Workflows immediately and, if the limit is reached, by marking the remaining Workflows and the Workflow association for deferred deletion by proc_AutoCleanupWorkflows.

Return Code Values: An integer which MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.4.30 proc_DropWorkItem

The proc_DropWorkItem stored procedure is called to delete an existing Work Item from the set of pending Work Items for a Site Collection. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_DropWorkItem(
    @SiteId            uniqueidentifier,
    @Id                uniqueidentifier
);
```

@SiteId: The Site Collection identifier of the Site Collection of the Work Item. MUST NOT be NULL.

@Id: The Work Item identifier. The server MUST only delete the Work Item associated with this Work Item identifier. MUST NOT be NULL.

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

Result Sets: MUST NOT return any result sets.

3.1.4.31 proc_EnableDeclarativeWorkflowAssociations

The proc_EnableDeclarativeWorkflowAssociations stored procedure is called to enable or disable all Declarative Workflow associations contained in a Site Collection. When a Workflow association is disabled, no new Workflows can be created based on that Workflow association. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE dbo.proc_EnableDeclarativeWorkflowAssociations(
    @SiteId                  uniqueidentifier,
    @Enabled                 int
);
```

@SiteId: The Site Collection identifier of the Site Collection which contains the Workflow associations.

@Enabled: This parameter determines whether the Workflow associations are enabled or disabled. This value MUST be 0 or 1. When set to 1, the server MUST enable all Declarative Workflow associations in the Site Collection. When set the 0, the server MUST disable all the Declarative Workflow associations in the Site Collection, and MUST NOT allow any new Workflows to be created from the disabled Workflow associations.

Return Code Values: An integer which MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.4.32 proc_EnsureWorkflowStatusFieldValue

The proc_EnsureWorkflowStatusFieldValue stored procedure is called to write the workflow template status field and add it to a list Item. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE dbo.proc_EnsureWorkflowStatusFieldValue (
    @TemplateId              uniqueidentifier,
    @SiteId                  uniqueidentifier,
    @WebId                   uniqueidentifier,
    @ListId                  uniqueidentifier,
    @ItemId                  int,
    @Level                   tinyint,
    @Version                 int,
    @StatusFieldOrdinal      int,
    @StatusField             nvarchar(64)
);
```

@TemplateId: The Workflow Template Identifier of the workflow template on which the workflow status field is being added. There is one status field for a given workflow template. MUST NOT be NULL.

@SiteId: The site collection identifier of the site collection which contains the workflow. MUST NOT be NULL.

@WebId: The site identifier, as specified in [MS-WSSFO], section 2.2.1.10, of the site which contains the workflow. MUST NOT be NULL.

@ListId: The list identifier of the list which contains the workflow. MUST NOT be NULL.

@ItemId: The list item identifier (as specified in [MS-WSSFO], section 2.2.1.6) of the list item the workflow status is created for. MUST NOT be NULL.

@Level: If @ListId represents a document library, this represents the publishing level of the document corresponding to the list item specified by @ItemId. If @List does not represent a document library, this MUST be 1.

@Version: The list item version of the list item, or the value 0.

@StatusFieldOrdinal: The ordinal of the workflow Status field of the workflow. MUST NOT be NULL.

@StatusField: The name of the workflow status field of the workflow. The server MUST update the field specified by @StatusField and @StatusFieldOrdinal of the list item specified by @ItemId to the workflow identifier of the workflow. MUST NOT be NULL.

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

Value	Description
0	Successful completion.
87	At least one input parameter was invalid.

Result Sets: MUST NOT return any result sets.

3.1.4.33 proc_EnumerateWebPartsForList

The proc_EnumerateWebPartsForList stored procedure is called to return properties of Web Parts in shared views from Published Web Part Pages contained within the specified list. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_EnumerateWebPartsForList(
    @SiteId                 uniqueidentifier,
    @WebId                  uniqueidentifier,
    @ListId                 uniqueidentifier
);
```

@SiteId: The Site Collection identifier for the Site Collection which contains the list.

@WebId: The Site identifier (as specified in [MS-WSSFO], section 2.2.1.10) of the Site which contains the list.

@ListId: The list identifier (as specified in [MS-WSSFO], section 2.2.1.5) of the list.

Return Code Values: An integer which MUST be 0.

Result Sets: MUST return the following result set:

3.1.4.33.1 Web Parts Result Set

The Result Set is defined in section 2.2.6.2.

3.1.4.34 proc_EnumerateWebPartsForWeb

The proc_EnumerateWebPartsForWeb stored procedure is called to return properties of Web Parts in Shared Views of the specified Site. Only Web Parts from Published Web Part Pages are returned. The T-SQL syntax for the stored procedure is as follows:

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

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

@WebId: The Site identifier, as specified in [MS-WSSFO], section 2.2.1.10, of the Site.

Return Code Values: An integer which MUST be 0.

Result Sets: MUST return the following result set:

3.1.4.34.1 Web Parts Result Set

The Result Set is defined in section 2.2.6.2.

3.1.4.35 proc_FailOverInProgressWorkItems

The proc_FailOverInProgressWorkItems stored procedure is called to mark a set of Work Items as not In Progress Work Items. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_FailOverInProgressWorkItems(
    @ProcessingId             uniqueidentifier
);
```

@ProcessingId: The Work Item Process identifier of the Work Item Process. This parameter MUST NOT be NULL. For each Work Item associated with the given Work Item Process identifier for which the Work Item Delivery Date has passed, the server MUST do the following:

- Mark the Work Item as not In Progress Work Items.
- Set the Work Item Process identifier of the Work Item to NULL.

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

Result Sets: MUST NOT return any result sets.

3.1.4.36 proc_GetAllWebPartsOnPage

The proc_GetAllWebPartsOnPage stored procedure is called to return information about all of the Web Parts on a Web Part Page. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetAllWebPartsOnPage(
    @SiteId                   uniqueidentifier,
    @CurrentWebId             uniqueidentifier,
    @AllUsers                 bit,
    @SystemID                 tSystemID,
    @DirName                  nvarchar(256),
    @LeafName                 nvarchar(128),
    @Level                    tinyint,
    @PrefetchListScope        bit,
    @GetViewBodies            bit
);
```

@SiteId: The Site Collection identifier of the site collection which contains the Web Part Page from which to get Web Parts.

@CurrentWebId: The identifier of the Site containing the Web Part Page from which to get Web Parts.

@AllUsers: Specifies whether to return Web Parts for the Shared View or personal View of the Web Part Page. If set to 1, Web Parts for the Shared View are returned in the [MS-WSSFO2].proc_FetchDocForHttpGet.Web Parts Metadata, Non-Personalized Result Set (as specified in [MS-WSSFO2] section 3.1.5.19.18). If set to 0, Web Parts personalized for the current user are returned in the [MS-WSSFO2].proc_FetchDocForHttpGet.WebParts Metadata, Personalized Result Set (as specified in [MS-WSSFO2], section 3.1.5.19.17).

@SystemID: The **SystemID** of the user originating the request or NULL to indicate an **anonymous user**. If @AllUsers is 0.

@DirName: The Directory Name of the Web Part Page.

@LeafName: The Leaf Name of the Web Part Page.

@Level: The Publishing Level of the Web Part Page from which to get Web Parts.

@PrefetchListScope: This value MUST be set to 1.

@GetViewBodies: This value MUST be set to 1.

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

Value	Description
0	Successful completion.
-2147467259	An error occurred while the stored procedure was running.
1	An error occurred while the stored procedure was running.
2	Specified Web Part Page does not exist.

Result Sets: MUST return zero, two or four result sets. No result set is returned when this stored procedure does not complete successfully. If two or four result sets are returned, first, either the Web Parts Metadata, Non-Personalized or the Web Parts Metadata, Personalized Result Set MUST be returned. Second, the list Metadata Result Set MUST be returned. If the list Metadata Result Set is empty then the rest of the result sets MUST NOT be returned.

3.1.4.36.1 Web Parts Metadata, Non-Personalized Result Set

If @AllUsers is 1, [MS-WSSFO2].proc_FetchDocForHttpGet.Web Parts Metadata, Non-Personalized Result Set MUST be returned (as specified in [MS-WSSFO2] section 3.1.5.19.18).

3.1.4.36.2 Web Parts Metadata, Personalized Result Set

If @AllUsers is 0, [MS-WSSFO2].proc_FetchDocForHttpGet.Web Parts Metadata, Personalized Result Set MUST be returned (as specified in [MS-WSSFO2], section 3.1.5.19.17).

3.1.4.36.3 List Metadata, Result Set

[MS-WSSFO2].proc_FetchDocForHttpGet.List Metadata, Result Set MUST be returned (as specified in [MS-WSSFO2] section 3.1.5.19.19).

3.1.4.36.4 List Event Receivers, Result Set

If List Metadata, Result Set is NOT empty then [MS-WSSFO2].proc_FetchDocForHttpGet.List Event Receivers, Result Set MUST be returned (as specified in [MS-WSSFO2] section 3.1.5.19.20).

3.1.4.36.5 List Security Information, Result Set

If List Metadata, Result Set is NOT empty then [MS-WSSFO2].proc_FetchDocForHttpGet.List Security Information, Result Set MUST be returned (as specified in [MS-WSSFO2] section 3.1.5.19.21).

3.1.4.37 proc_GetContextCollectionEventReceivers

The proc_GetContextCollectionEventReceivers stored procedure is called to retrieve, for a specific Site Collection and Context Collection, a collection of event receivers of a specific Context type. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetContextCollectionEventReceivers(
    @SiteId                 uniqueidentifier,
    @ContextCollectionId    uniqueidentifier,
    @ContextType            uniqueidentifier = NULL
);
```

@SiteId: The Site Collection identifier of the site collection for which the event receivers are requested to be returned. This value MUST NOT be NULL.

@ContextCollectionId: The context collection identifier for the Context Collection for which the event receivers are requested to be returned.

@ContextType: The context type identifier for the Context type the event receivers have to match. The default is NULL. When this value is NULL, event receivers of any Context type are returned.

Return Code Values: An integer which MUST be 0.

Result Sets: MUST return the following result set:

3.1.4.37.1 Event Receivers Result Set

The Result Set is defined in the Event Receivers Result Set (as specified in [MS-WSSFO], section 2.2.5.9).

3.1.4.38 proc_GetContextObjectEventReceivers

The proc_GetContextObjectEventReceivers stored procedure is called to retrieve a list of Event Receivers and optionally remove an Event Receiver. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetContextObjectEventReceivers(
    @SiteId                 uniqueidentifier,
    @ContextObjectId        uniqueidentifier,
    @DeleteHostLookupId     uniqueidentifier = NULL,
    @HostType               int = NULL
);
```

@SiteId: The Site Collection identifier of the site collection which contains the Site for which the Event Receivers have to be requested. This value MUST NOT be NULL.

@ContextObjectId: The context object identifier for the Context Object of the Workflow receiver process associated with the Event Receivers to be requested. This value MUST NOT be NULL.

@DeleteHostLookupId: The Event Receiver identifier (as specified in [MS-WSSFO], section 2.2.1.3) for the Event Receiver to be optionally removed. The value MAY be NULL if no Event Receivers need to be deleted. If the value is NOT NULL, the Event Receiver, with Site Collection identifier property equal to @SiteId and Event Receiver identifier (as specified in [MS-WSSFO], section 2.2.1.3) equal to @DeleteHostLookupId, will be deleted. The default is NULL.

@HostType: The Event Host type (as specified in [MS-WSSFO], section 2.2.3.5) of the Event Receivers that are requested. If this parameter is NOT NULL, the results are filtered for HostType=@HostType. If this parameter is NULL, no result filtering is performed and Event Receivers with any value of Event Host type (as specified in [MS-WSSFO], section 2.2.3.5) are returned. The default is NULL.

Return Code Values: An integer which MUST be 0.

Result Sets: MUST return the following result set when (a) @DeleteHostLookupId is set to NULL OR (b) DeleteHostLookupId is NOT NULL AND the Event Receiver it refers to exists AND its type is equal to ContextLookupReceivers Event Receiver type (as specified in [MS-WSSFO], section 2.2.3.6). In the case where @DeleteHostLookupId is NOT NULL and the Event Receiver it refers to does not exist OR its type is not equal to ContextLookupReceivers Event Receiver type (as specified in [MS-WSSFO], section 2.2.3.6), the stored procedure MUST NOT return a result set.

3.1.4.38.1 Event Receivers Result Set

This Result Set will be filtered by the Site through @SiteId and the context object identifier through @**ContextObjectId**, which MUST both be specified. The optional parameter @HostType can be used to further filter the results. Note that when @HostType is set to NULL, no HostType filtering is performed and all rows with any value of HostType are returned. The Result Set is defined in the Event Receivers Result Set (as specified in [MS-WSSFO], section 2.2.5.9).

3.1.4.39 proc_GetDocEventReceivers

The proc_GetDocEventReceivers stored procedure is called to read all event receivers registered for a specified document. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetDocEventReceivers(
    @DocSiteId              uniqueidentifier,
    @DocWebId               uniqueidentifier,
    @DocUrl                 nvarchar(260)
);
```

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

@DocWebId: The Site identifier (as specified in [MS-WSSFO], section 2.2.1.10) of the site which contains the document.

@DocUrl: The store-relative form URL of the document.

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

Value	Description
0	Successful completion.
3	The document does not exist.

Result Sets: MUST return the following result set that contains one row for each of the event receivers registered for the specified document when the return code is 0. proc_GetDocEventReceivers MUST NOT return a result set when the return code is NOT 0.

3.1.4.39.1 Event Receivers Result Set

The Result Set is defined in the Event Receivers Result Set (as specified in [MS-WSSFO], section 2.2.5.9).

3.1.4.40 proc_GetListItemWorkflows

The proc_GetListItemWorkflows stored procedure is called to obtain a set of Workflows. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE dbo.proc_GetListItemWorkflows(
    @SiteId                 uniqueidentifier,
    @WebId                  uniqueidentifier,
    @ListId                 uniqueidentifier,
    @ItemId                 int,
    @WorkflowInstanceId     uniqueidentifier,
    @TemplateId             uniqueidentifier,
    @InclusiveFilterState   int = 0xFFFFFFFF,
    @ExclusiveFilterState   int = 0
);
```

@SiteId: The Site Collection identifier of the Site Collection which contains the Workflows.

@WebId: The Site identifier (as specified in [MS-WSSFO], section 2.2.1.10) of the Site which contains the Workflows.

@ListId: The list identifier of the list which contains the list Items the Workflows were created for. If this value is NULL, the server MUST include all lists.

@ItemId: The list Item identifier (as specified in [MS-WSSFO], section 2.2.1.6) of the list Items the Workflows were created for. If @ListId is NULL, this value MUST be NULL. If this value is NULL, the server MUST include all list Items.

@WorkflowInstanceId: The Workflow identifier of the Workflow. If this value is NULL, the server MUST include all Workflows. If this value is not NULL, the server MUST ignore @WebId, @ListId, @ItemId and @TemplateId and return only one row in the Result Set which contains the Workflow specified by @WorkflowInstanceId.

@TemplateId: The Workflow Template identifier of the workflow template. If this value is NULL, the server MUST include all workflow templates.

@InclusiveFilterState: A workflow internal state bitmask. The server MUST include only Workflows that have at least one internal state flag in common with the bitmask (that is, Workflow.InternalState & @InclusiveFilter <> 0).

@ExclusiveFilterState: A workflow internal state bitmask. The server MUST exclude all Workflows that have any internal state flags in common with the bitmask (that is, Workflow.InternalState & ExclusiveFilter <> 0).

Return Code Values: An integer which MUST be 0.

Result Sets: MUST return the list Item Workflows Result Set specified in section 2.2.6.1. The InstanceData and ProcessingId columns in the Result Set MUST be NULL and the InstanceDataSize column MUST contain the value 0.

3.1.4.40.1 List Item Workflows Result Set

The Result Set is defined in section 2.2.6.1.

3.1.4.41 proc_GetListItemWorkflowWithInstanceDataAndLock

The proc_GetListItemWorkflowWithInstanceDataAndLock stored procedure is called to lock a workflow and get back a result set for the workflow. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetListItemWorkflowWithInstanceDataAndLock(
    @SiteId                 uniqueidentifier,
    @WebId                  uniqueidentifier,
    @ListId                 uniqueidentifier,
    @ItemId                 int,
    @WorkflowInstanceId     uniqueidentifier,
    @HasInstanceData        int OUTPUT
);
```

@SiteId: The Site Collection identifier of the Site Collection which contains the Workflow.

@WebId: The Site identifier (as specified in [MS-WSSFO], section 2.2.1.10) of the Site which contains the Workflow.

@ListId: The protocol server MUST ignore this parameter.

@ItemId: The protocol server MUST ignore this parameter.

@WorkflowInstanceId: The Workflow identifier of the Workflow. The server MUST attempt to lock the Workflow.

@HasInstanceData: The server MUST ignore the input value of this parameter. If the server locked the Workflow, the server MUST set the output value to 1. If the server did not lock the workflow, the server MUST set the output value to 0.

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

Value	Description
0	Successful completion.
5	Error: Access denied.
19	Error: The workflow was not found or was locked.
82	Error: Failed to lock the workflow.

Result Sets: MUST return the List Item Workflows Result Set with exactly one row containing the Workflow specified by @WorkflowInstanceId. If the Workflow was successfully locked, the InstanceData, InstanceDataSize and ProcessingId columns MUST contain the instance data for the Workflow, the instance data size, and the identifier of the computer processing the Workflow, respectively. If the workflow was not successfully locked, the InstanceData and ProcessingId columns MUST be null and the InstanceDataSize column MUST contain the value 0.

3.1.4.42 proc_GetListWebParts

The proc_GetListWebParts stored procedure is called to return a Result Set of list view Web Parts, list form Web Parts, **Data View Web Parts**, and Data Form Web Parts associated with the specified list in Web Part pages. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetListWebParts(
    @ListId                 uniqueidentifier,
    @UserID                 int,
    @bGetAllLevel           bit,
    @bGetDeleted            bit = 0,
    @bGetAllUsers           bit = 0,
    @SiteId                 uniqueidentifier
);
```

@ListId: The list identifier (as specified in [MS-WSSFO], section 2.2.1.5).

@UserID: The User identifier (as specified in [MS-WSSFO], section 2.2.1.12) of the current user.

@bGetAllLevel: The parameter determines whether to include Web Parts from Web Part Pages with all Publishing Levels. When this parameter is set to 1, the Result Set MUST include Web Parts from Web Part Pages with all Publishing Levels. When set to 0, the Result Set MUST only include Web Parts from Web Part Pages with the highest Publishing Level that the current user has permission to view.

@bGetDeleted: The parameter determines whether to include Web Parts from Web Part Pages that are in the **Recycle Bin**. When set to 1, the Result Set MUST include Web Parts that are in Web Part Pages that are in the Recycle Bin. When set to 0, the Result Set MUST only return Web Parts that are in Web Part Pages that are not in the Recycle Bin.

@bGetAllUsers: The parameter determines whether to include Web Parts for All Users, or just the current user. When set to 1, the returned Result Set MUST return Web Parts for All Users, including Web Parts in other user's personal views. When set to 0, the returned Result Set MUST only return Web Parts in Shared Views or personal Views of current user.

@SiteId: The Site Collection identifier of the Site Collection which contains the specified list.

Return Code Values: An integer which MUST be 0.

Result Sets: MUST return the following result set:

3.1.4.42.1 List Web Parts Result Set

This Result Set returns Web Part information associated with the list in Web Part pages, one row per Web Part, ordered by the time the Web Part was added to the Web Part Page. The T-SQL syntax for the result set is as follows:

```
tp_ListId               uniqueidentifier,
tp_Type                 tinyint,
tp_Id                   uniqueidentifier,
tp_Flags                int,
tp_DisplayName          nvarchar(255),
tp_PageUrl              nvarchar(260),
tp_BaseViewId           tinyint,
tp_View                 ntext,
tp_Level                tinyint,
```

```
tp_ContentTypeId        varbinary(512);
```

tp_ListId: The list identifier (as specified in [MS-WSSFO], section 2.2.1.5) of the list that contains the Web Part. This MUST be the same as the @ListId parameter.

tp_Type: The Page type (as specified [MS-WSSFO2] section 2.2.3.14) of the Web Part.

tp_Id: The GUID that identifies the Web Part. This value MUST NOT be NULL.

tp_Flags: The View Flags (as specified in [MS-WSSFO], section 2.2.2.11) of the Web Part.

tp_DisplayName: The Display Name of the Web Part.

tp_PageUrl: The URL of the Web Part Page for the Web Part, in Store-Relative Form.

tp_BaseViewId: The base view identifier for the Web Part.

tp_View: The CAML of the Web Part**.**

tp_Level: The Publishing Level of the Web Part Page.

tp_ContentTypeId: The Content type identifier (as specified in [MS-WSSFO], section 2.2.1.1). If the Web Part is a list View Web Part, returns the Content type identifier (as specified in [MS-WSSFO], section 2.2.1.1) of the Content type associated with this view. If the Web Part is not associated with any Content type, then it MUST return 0x. It MUST NOT be NULL.

3.1.4.43 proc_GetNextWebPartOrder

The proc_GetNextWebPartOrder stored procedure is called to request a Web Part Zone Index that is one larger than the maximum Web Part Zone Index being used by all of the Web Parts in a Web Part Zone. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetNextWebPartOrder(
    @SiteID                 uniqueidentifier,
    @DocID                  uniqueidentifier,
    @ZoneId                 nvarchar(64),
    @NextOrder              int OUTPUT
);
```

@SiteID: The Site Collection identifier of the site collection which contains the Web Part Page.

@DocID: The Document identifier (as specified in [MS-WSSFO], section 2.2.1.2) of the Web Part Page which contains the Web Part Zone.

@ZoneId: The Web Part Zone identifier of the Web Part Zone to calculate the next Web Part Zone Index for.

@NextOrder: A Web Part Zone Index that is one larger than the maximum Web Part Zone Index present in the Web Part Zone, returned as an output parameter. This value MUST be 1 if no Web Part zone indexes are present in the Web Part Zone or if @SiteID, @DocID do not reference an existing Web Part Page or @ZoneId does not reference an existing Web Part Zone on the Web Part Page.

Return Code Values: An integer which MUST be 0. Output parameter @NextOrder MUST return calculated Web Part Zone Index value.

Result Sets: MUST NOT return any result sets.

3.1.4.44 proc_GetRecycleBinItemEventReceivers

The proc_GetRecycleBinItemEventReceivers stored procedure is called to read the information and event receivers of a specified **Recycle Bin item**. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetRecycleBinItemEventReceivers(
    @SiteId                 uniqueidentifier,
    @WebId                  uniqueidentifier,
    @UserId                 int,
    @DeleteTransactionId    varbinary(16)
);
```

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

@WebId: The Site identifier (as specified in [MS-WSSFO], section 2.2.1.10) of the site which contains the recycle bin item.

@UserId: The User identifier (as specified in [MS-WSSFO], section 2.2.1.12) of the current user.

@DeleteTransactionId: The **delete transaction identifier** of the recycle bin item.

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

Value	Description
0	Successful completion.
1168	No recycle bin item is found for @SiteId AND @DeleteTransactionID when @UserId is 0; OR no recycle bin item is found for @SiteId, @WebId, @DeleteTransactionID AND @UserId when @UserId is NOT 0; OR more than one recycle bin item is found for the given parameters.

Result Sets: MUST return three result sets in the following order when the return code is 0 AND MUST NOT return any result sets when the return code is NOT 0.

3.1.4.44.1 Recycle Bin Item Result Set

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

```
ItemType                    tinyint,
WebUrl                      nvarchar(256),
ListId                      uniqueidentifier,
ListTitle                   nvarchar(255),
ListItemId                  int,
DocId                       uniqueidentifier;
```

ItemType: The type of the recycle bin item. The value MUST be one of the following.

Value	Description
1	Recycle bin item is a document.

Value	Description
2	Recycle bin item is a **document version**.
3	Recycle bin item is a list item.
4	Recycle bin item is a list.
5	Recycle bin item is a **folder**.
6	Recycle bin item is a folder with lists.
7	Recycle bin item is an **attachment**.
8	Recycle bin item is a version of a list item.

WebUrl: The URL in store-relative form of the site that contained the recycle bin item.

ListId: The list identifier (as specified in [MS-WSSFO], section 2.2.1.5) of the **Recycle Bin item list**.

ListTitle: The title of the recycle bin item list.

ListItemId: The list Item identifier (as specified in [MS-WSSFO], section 2.2.1.6) corresponding to the recycle bin item when the recycle bin item type is 1, 3, 5, 7, or 8. Otherwise ListItemId MUST be NULL when the recycle bin item type is 2, 4, or 6.

DocId: The Document identifier (as specified in [MS-WSSFO], section 2.2.1.2) when the recycle bin item has a corresponding document. Otherwise, DocId is NULL.

3.1.4.44.2 List Event Receivers Result Set

This result set contains all the event receivers of the recycle bin item list. The **result set** is defined in the Event Receivers Result Set (as specified in [MS-WSSFO], section 2.2.5.9).'

3.1.4.44.3 Site Event Receivers Result Set

This result set contains all the event receivers of the Site that contained the recycle bin item. The **result set** is defined in the Event Receivers Result Set (as specified in [MS-WSSFO], section 2.2.5.9).'

3.1.4.45 proc_GetRunnableWorkItems

The proc_GetRunnableWorkItems stored procedure is called to retrieve a restricted set of Work Items for which the Work Item Delivery Date has passed and mark them as In Progress Work Item. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetRunnableWorkItems(
    @ProcessingId             uniqueidentifier,
    @SiteId                   uniqueidentifier,
    @WorkItemType             uniqueidentifier,
    @BatchId                  uniqueidentifier,
    @MaxFetchSize             int= 1000,
    @ThrottleThreshold        int= 0
);
```

@ProcessingId: The Work Item Process identifier of the Work Item Process. The server MUST set to this value the Work Item Process identifier of any Work Items that it modifies. MUST NOT be NULL.

@SiteId: The Site Collection identifier of the Site Collection. If the parameter is not NULL, then the server MUST only modify and return Work Items associated with this Site Collection. If the parameter is NULL, then the server MUST modify and return Work Items that meet the criteria specified by the other parameters regardless of associated Site Collection.

@WorkItemType: The Work Item type identifier of the Work Item type. The server MUST only modify and return Work Items associated with this Work Item type. MUST NOT be NULL.

@BatchId: The Work Item Batch identifier of the Work Item Batch. If the parameter is not NULL, then the server MUST only modify and return Work Items associated with this Work Item Batch identifier and MUST also mark those Work Items as Throttled Fetch. If the parameter is NULL, then the server MUST modify and return Work Items that meet the criteria specified by the other parameters regardless of associated Work Item Batch identifier.

@MaxFetchSize: The maximum number of Work Items that will be marked as In Progress Work Items. This parameter MUST be non-negative. If the value of the parameter is not 0, then the server MUST limit to the specified value the number of new Work Items it marks as in progress | In Progress Work Item. If the value of the parameter is 0, then the server MUST NOT limit the number of items it modifies based on this parameter. MUST NOT be NULL.

@ThrottleThreshold: A limit on the number of work item batches. This parameter MUST be non-negative. This parameter MUST NOT be NULL. The server MUST NOT mark any new items as In Progress Work Item if:

- The value of this parameter is not 0.
- The value of this parameter is less than the number of distinct work item batch identifiers in the set of Work Items matching the following criteria:
  - Is marked as In Progress Work Item,
  - Is marked as Throttled Fetch,
  - Has an associated Work Item type is given by @WorkItemType, and
  - Has a Work Item Delivery Date that has passed.

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

Value	Description
0	Successful completion.
5	Error: Access denied.

Result Sets: MUST return zero or one result sets:

3.1.4.45.1 Work Items Result Set

This Result Set returns the Work Items that are marked as In Progress Work Items and match the criteria specified by the parameters. The Result Set is defined in section 2.2.6.3.

3.1.4.46 proc_GetWorkflowAssociations

The proc_GetWorkflowAssociations stored procedure is called to get a set of Workflow associations. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetWorkflowAssociations (
    @SiteId                 uniqueidentifier,
    @WebId                  uniqueidentifier,
    @Id                     uniqueidentifier,
    @ListId                 uniqueidentifier,
    @ContentTypeId          varbinary(512)
);
```

@SiteId: The Site Collection identifier of the Site Collection which contains the Workflow associations.

@WebId: The Site identifier (as specified in [MS-WSSFO], section 2.2.1.10) of the Site which contains the Workflow associations. If this value is NULL, the server MUST include all Sites.

@Id: The Workflow association identifier of the Workflow association. If this value is not null, the server MUST ignore @WebId, @ListId and @ContentTypeId, and the Result Set MUST contain exactly one row containing the Workflow association specified by @Id.

@ListId: The list identifier of the list the Workflow associations are associated with. If this value is NULL, the server MUST include all lists.

@ContentTypeId: The Content type identifier (as specified in [MS-WSSFO], section 2.2.1.1) of the Content type the Workflow associations are associated with. If this value is NULL, the server MUST include all Content types.

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

Result Sets: MUST return the Workflow Associations Result Set.

3.1.4.46.1 Workflow Associations Result Set

The Result Set is defined in section 2.2.6.3.

3.1.4.47 proc_GetWorkflowDataForItem

The proc_GetWorkflowDataForItem stored procedure is called to obtain data about Workflows and Workflow associations. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetWorkflowDataForItem(
    @SiteId                 uniqueidentifier,
    @WebId                  uniqueidentifier,
    @ListId                 uniqueidentifier,
    @ItemId                 int,
    @ContentTypeId          varbinary(512),
    @gwfdi                  int = 0xF,
    @InclusiveFilterState   int = 0xFFFFFFFF,
    @ExclusiveFilterState   int = 0
);
```

@SiteId: The Site Collection identifier of the Site Collection which contains the Workflows and Workflow associations.

@WebId: The Site identifier (as specified in [MS-WSSFO], section 2.2.1.10) of the Site which contains the Workflows and Workflow associations.

@ListId: The list identifier of the list which contains the list Items the Workflows were created for. If this value is NULL, the server MUST include all lists.

@ItemId: The list Item identifier (as specified in [MS-WSSFO], section 2.2.1.6) of the list Item the Workflows were created for. If @ListId is NULL, @ItemId MUST be NULL. If this value is NULL, the server MUST include all lists.

@ContentTypeId: The Content type identifier (as specified in [MS-WSSFO], section 2.2.1.1) with which the Workflows are associated.

@gwfdi: A bitmask which determines which Result Sets are returned. MUST contain zero or more of the flags listed in the following table:

Value	Description
1	The server MUST return the Workflow Associations Result Set for the list specified by @ListId. @ListId MUST NOT be NULL.
2	The server MUST return the Workflow Associations Result Set for the Content type specified by @ContentTypeId. @ContentTypeId MUST NOT be NULL.
4	The server MUST return the List Item Workflows Result Set for the list Item specified by @ItemId. @ListId and @ItemId MUST NOT be NULL.

@InclusiveFilterState: A workflow internal state bitmask. The server MUST include only Workflows that have at least one internal state bit flag in common with @InclusiveFilterState (that is, Workflow.InternalState & @InclusiveFilterState <> 0) in the List Item Workflows Result Set.

@ExclusiveFilterState: A workflow internal state bitmask. The server MUST exclude any Workflows that have any internal state bit flags in common with @ExclusiveFilterState (that is, Workflow.InternalState & @ExclusiveFilterState <> 0) from the List Item Workflows Result Set.

Return Code Values: An integer which MUST be 0.

Result Set: MUST return zero, one, or two Workflow Associations Result Sets and 0 or 1 List Item Workflows Result Set based on the @gwfdi parameter, ordered from the lowest flag (1) to the highest (4).

3.1.4.47.1 Workflow Associations Result Set

The Result Set is defined in 2.2.6.3.

3.1.4.47.2 List Item Workflows Result Set

If the List Item Workflows Result Set is returned, the InstanceData and ProcessingId columns MUST be null and the InstanceDataSize column MUST contain the value 0. The Result Set is defined in section 2.2.6.1.

3.1.4.48 proc_GetWorkItems

The proc_GetWorkItems stored procedure is called to retrieve a set of Work Items that meet the specified criteria. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetWorkItems(
    @SiteId                 uniqueidentifier,
    @ParentId               uniqueidentifier,
    @WorkItemType           uniqueidentifier,
    @BatchId                uniqueidentifier,
    @WorkItemId             uniqueidentifier
);
```

@SiteId: The Site Collection identifier of the Site Collection. The server MUST only return Work Items associated with this Site Collection identifier. MUST NOT be NULL.

@ParentId: The Work Item Parent identifier of the Work Item. If @WorkItemId is NULL and this parameter is not NULL, then the server MUST only return Work Items which have this Work Item Parent identifier. If this parameter is NULL, then the server MUST return Work Items that meet the criteria specified by the other parameters, regardless of the value of their Work Item Parent identifier.

@WorkItemType: The Work Item type identifier of the Work Item type. If @WorkItemId is NULL, then this parameter MUST NOT be NULL, and the server MUST only return Work Items associated with this Work Item type. If @WorkItemId is not NULL, then the server MUST return Work Items that meet the criteria specified by the other parameters, regardless of associated Work Item type.

@BatchId: The Work Item Batch identifier of the work item batch. If @WorkItemId is NULL and the parameter is not NULL, then the server MUST only return Work Items associated with this Work Item Batch. If this parameter is NULL, then the server MUST return Work Items that meet the criteria specified by the other parameters, regardless of the associated work item batch.

@WorkItemId: The work item identifier. If the parameter is not NULL, then the server MUST restrict the returned Work Item to have a Work Item identifier matching the parameter and associated with the Site Collection indicated by @SiteId. If this parameter is NULL, then the server MUST return Work Items that meet the criteria specified by the other parameters, regardless of associated work item identifier.

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

Result Sets: MUST return the following result set:

3.1.4.48.1 Single Work Item Result Set

This Result Set returns a single Work Item that meets the criteria specified by the parameters when @WorkItemId is not NULL. The Result Set is defined in section 2.2.6.3.

3.1.4.48.2 Multiple Work Items Result Set

This Result Set returns the Work Items that meet the criteria specified by the parameters when @WorkItemId is NULL. The Result Set is defined in section 2.2.6.3.

3.1.4.49 proc_InsertContextEventReceiver

The proc_ InsertContextEventReceiver stored procedure is called to create a new Event Receiver and, optionally, create an additional Event Receiver that the new Event Receiver will be registered against. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_InsertContextEventReceiver(
    @Id                         uniqueidentifier,
    @Name                       nvarchar(256),
    @SiteId                     uniqueidentifier,
    @WebId                      uniqueidentifier,
    @ParentHostId               uniqueidentifier,
    @ParentHostType             int,
    @Type                       int,
    @SequenceNumber             int,
    @Assembly                   nvarchar(256),
    @Class                      nvarchar(256),
    @Data                       nvarchar(256),
    @Filter                     nvarchar(256),
    @Credential                 int,
    @ContextHostType            int,
    @ContextObjectItemId        int,
    @ContextObjectUrl           nvarchar(260),
    @ContextType                uniqueidentifier,
    @ContextEventType           uniqueidentifier,
    @ContextId                  uniqueidentifier,
    @ContextObjectId            uniqueidentifier,
    @ContextCollectionId        uniqueidentifier
);
```

@Id: The Event Receiver identifier (as specified in [MS-WSSFO], section 2.2.1.3) of the Event Receiver. This value MUST NOT be NULL.

@Name: The name of the Event Receiver. This value MUST NOT be NULL.

@SiteId: The Site Collection identifier of the Site Collection which contains the Event Host. This value MUST NOT be NULL.

@WebId: The Site identifier (as specified in [MS-WSSFO], section 2.2.1.10) of the Site which contains the Event Host. This value MUST NOT be NULL.

@ParentHostId: The Event Host identifier of the Event Host with which the Event Receiver is associated. This parameter MUST NOT be NULL.

@ParentHostType: The type of the Event Host with which the Event Receiver is associated. @ParentHostType MUST be a value of the Event Host type (as specified in [MS-WSSFO], section 2.2.3.5).

@Type: The type of the Event Receiver. @Type MUST be a value of the Event Receiver type (as specified in [MS-WSSFO], section 2.2.3.6).

@SequenceNumber: The sequence number (1) of the event receiver. @SequenceNumber MUST be greater than OR equal to zero AND less than OR equal to 65535.

@Assembly: The Assembly Name strong name of the **assembly** that contains the Event Receiver. This value MUST NOT be NULL.

@Class: The fully qualified class name of the Event Receiver in the assembly. This value MUST NOT be NULL.

@Data: Additional data to be passed to the Event Receiver.

@Filter: Reserved. @Filter MUST be NULL.

@Credential: Reserved. @Credential MUST be zero.

@ContextHostType: The type of the event host of the event receiver. The value MUST be one of Event Host type (as specified in [MS-WSSFO], section 2.2.3.5).

@ContextObjectItemId: The context object identifier of the Event Host for which an Event Receiver is registered.

@ContextObjectUrl: Reserved. @ContextObjectUrl MUST be NULL.

@ContextType: The context type identifier of the event receiver.

@ContextEventType: Reserved. @ContextEventType MUST be NULL.

@ContextId: The context identifier of the event receiver.

@ContextObjectId: The context object identifier for the Event Host of the event receiver.

@ContextCollectionId: The context collection identifier of the event receiver.

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

Value	Description
0	Successful completion.
87	**@ContextCollectionId** is NULL and no Event Receivers were inserted OR the insertion of Event Receivers failed.

Result Sets: MUST NOT return any result sets.

3.1.4.50 proc_InsertDocEventReceiver

The proc_InsertDocEventReceiver stored procedure is called to register an event receiver for a specified document. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_InsertDocEventReceiver(
    @DocUrl                  nvarchar(260),
    @Id                      uniqueidentifier,
    @Name                    nvarchar(256),
    @SiteId                  uniqueidentifier,
    @WebId                   uniqueidentifier,
    @ItemId                  int,
    @Type                    int,
    @SequenceNumber          int,
    @Assembly                nvarchar(256),
    @Class                   nvarchar(256),
    @Data                    nvarchar(256),
    @Filter                  nvarchar(256),
    @Credential              int
```

```
);
```

@DocUrl: The URL in store-relative form of the document.

@Id: The Event Receiver identifier (as specified in [MS-WSSFO], section 2.2.1.3) of the event receiver. This value MUST NOT be NULL.

@Name: The name of the event receiver. This value MUST NOT be NULL.

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

@WebId: The Site identifier (as specified in [MS-WSSFO], section 2.2.1.10) of the site which contains the document.

@ItemId: Reserved. @ItemId MUST be zero.

@Type: The type of the event receiver. @Type MUST be one of Event Receiver type (as specified in [MS-WSSFO], section 2.2.3.6).

@SequenceNumber: The sequence number (1) of the event receiver. @SequenceNumber MUST be greater than OR equal to zero AND less than OR equal to 65535.

@Assembly: The assembly name of the implementation of the event receiver. This value MUST NOT be NULL.

@Class: The fully qualified class name of the implementation of the event receiver. This value MUST NOT be NULL.

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

@Filter: Reserved. @Filter MUST be NULL.

@Credential: Reserved. @Credential MUST be zero.

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

Value	Description
0	Insertion succeeded.
3	The document identified by @DocUrl is not found in the site identified by @WebId in the site collection identified by @SiteId.
87	The insertion failed.

Result Sets: MUST NOT return any result sets.

3.1.4.51 proc_InsertEventReceiver

The proc_InsertEventReceiver stored procedure is called to register an event receiver for a specified event host. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_InsertEventReceiver(
    @Id                         uniqueidentifier,
    @Name                       nvarchar(256),
```

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

@Id: The Event Receiver identifier (as specified in [MS-WSSFO], section 2.2.1.3) of the event receiver. This value MUST NOT be NULL.

@Name: The name of the event receiver. This value MUST NOT be NULL.

@SiteId: The Site Collection identifier of the site collection which contains the event host. This value MUST NOT be NULL.

@WebId: The Site identifier (as specified in [MS-WSSFO], section 2.2.1.10) of the site which contains the event host. This value MUST NOT be NULL.

@HostId: The event host identifier of the event host of the event receiver. This value MUST NOT be NULL.

@HostType: The type of the event host of the event receiver. @HostType MUST be one of Event Host type (as specified in [MS-WSSFO], section 2.2.3.5).

@ItemId: Reserved. @ItemId MUST be 0.

@DirName: Reserved. @DirName MUST be NULL.

@LeafName: Reserved. @LeafName MUST be NULL.

@Type: The type of the event receiver. @Type MUST be one of Event Receiver type (as specified in [MS-WSSFO], section 2.2.3.6).

@SequenceNumber: The sequence number (1) of the event receiver. @SequenceNumber MUST greater than OR equal to 0 AND less than OR equal to 65535.

@Assembly: The assembly name of the implementation of the event receiver. This value MUST NOT be NULL.

@Class: The fully qualified class name of the implementation of the event receiver. This value MUST NOT be NULL.

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

@Filter: Reserved. @Filter MUST be NULL.

@SourceId: The event receiver source identifier of the event receiver. This is the Feature identifier (as specified in [MS-WSSFO], section 2.2.1.4) of the feature if the event receiver is added via a feature. This is the Content type identifier (as specified in [MS-WSSFO], section 2.2.1.1) of the content type if the event receiver is added via a content type. Otherwise the event receiver source identifier MUST be NULL.

@SourceType: The Event Receiver Source type of the event receiver. @SourceType MUST be one of the Event Receiver Source Type values.

@Credential: Reserved. @Credential MUST be 0.

@ContextType: The context type identifier of the event receiver.

@ContextEventType: Reserved. @ContextEventType MUST be NULL.

@ContextId: The context identifier of the event receiver.

@ContextObjectId: The context object identifier for the Event Host of the event receiver.

@ContextCollectionId: The context collection identifier of the event receiver.

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

Value	Description
0	Insertion succeeded.
87	Insertion failed.

Result Sets: MUST NOT return any result sets.

3.1.4.52 proc_ProvisionWebPart

The proc_ProvisionWebPart stored procedure is called to add a new Web Part to a Web Part Page. If the Web Part is successfully added, its Web Part type identifier property MUST be set to NULL and the following Web Part properties MUST be set using the passed-in values: Publishing Level, IsIncluded, Frame State, Web Part Zone, Zone Index, and Source. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ProvisionWebPart(
    @SiteId                 uniqueidentifier,
    @DocID                  uniqueidentifier,
    @WebPartID              uniqueidentifier,
    @Level                  tinyint,
    @IsIncluded             bit,
    @FrameState             tinyint
    @ZoneID                 nvarchar(64),
    @PartOrder              int,
    @Source                 ntext
);
```

@SiteId: The Site Collection identifier of the Site Collection which contains the specified Web Part. MUST NOT be NULL.

@DocID: The Document identifier (as specified in [MS-WSSFO], section 2.2.1.2) of the Web Part Page where the Web Part is being added. MUST NOT be NULL.

@WebPartID: The Web Part identifier (as specified in [MS-WSSFO], section 2.2.1.14) of the Web Part within the Site Collection. MUST NOT be NULL.

@Level: The Publishing Level for the Web Part. MUST NOT be NULL.

@IsIncluded:. The Web Part Is Closed State for the Web Part. MUST NOT be NULL.

@FrameState: The Web Part Chrome State for the Web Part. MUST NOT be NULL.

@ZoneID: The Web Part Zone identifier of the Web Part Zone for the Web Part.

@PartOrder: The Web Part Zone Index for the Web Part.

@Source: The Web Part properties of the Web Part in either WPV2:WebPart format (as specified in [MS-WPPS], section 2.2.4.2) or HTML format.

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

Value	Description
0	Successful completion.
1	Adding the Web Part failed
212	The specified Site Collection is Locked..
1816	The Quota for the specified Site Collection has been exceeded.

Result Sets: MUST NOT return any result sets.

3.1.4.53 proc_RevertInProgressWorkItem

The **proc_RevertInProgressWorkItem** stored procedure is called to revert the Work Item specified by the parameters. Reverting a Work Item means to mark as no longer In Progress Work Item and possibly perform Exponential back-off on the Work Item Delivery Date; Exponential back-off only occurs if the Work Item in question is marked for Exponential Back-off. Before any reverts occur, however, proc_RevertInProgressWorkItem deletes the indicated Work Item if it is both 10 or more days past its Delivery Date and marked for automatic deletion. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_RevertInProgressWorkItem(
    @ProcessingId           uniqueidentifier,
    @SiteId                 uniqueidentifier,
    @Id                     uniqueidentifier
);
```

@ProcessingId: The Work Item Process identifier of the Work Item Process. The server MUST only consider for deletion and revert a Work Item if it is associated with the Work Item Process specified by this parameter. MUST NOT be NULL.

@SiteId: The Site Collection identifier of the Site Collection. The server MUST only consider for deletion and revert a Work Item if it is associated with the Site Collection specified by the parameter. MUST NOT be NULL.

@Id: The Work Item identifier. The server MUST only revert a Work Item if it has the specified Work Item identifier. MUST NOT be NULL.

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

Result Sets: MUST NOT return any result sets.

3.1.4.54 proc_RevertInProgressWorkItems

The **proc_RevertInProgressWorkItems** stored procedure is called to revert a set of Work Items specified by the parameters. Reverting a Work Item means to mark as no longer being In Progress Work Item and possibly perform Exponential back-off on the Work Item Delivery Date; Exponential back-off only occurs if the Work Item in question is marked for Exponential Back-off. Before any reverts occur, however, proc_RevertInProgressWorkItems deletes Work Items that are both 10 or more days past their Delivery Dates and marked for automatic deletion. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_RevertInProgressWorkItems(
    @ProcessingId             uniqueidentifier,
    @ProcessMachineId         uniqueidentifier,
    @SiteId                   uniqueidentifier,
    @ParentId                 uniqueidentifier,
    @WorkItemType             uniqueidentifier,
    @BatchId                  uniqueidentifier,
    @AnyRemaining             int OUTPUT
);
```

@ProcessingId: The Work Item Process identifier of the Work Item Process. The server MUST only consider for deletion and revert a Work Item if it is associated with the Work Item Process specified by this parameter. MUST NOT be NULL.

@ProcessMachineId: This parameter MUST be ignored.

@SiteId: The Site Collection identifier of the Site Collection. If the parameter is not NULL, then the server MUST only consider for deletion and revert Work Items associated with this Site Collection. If the parameter is NULL, then the server MUST operate on the Work Items specified by the other parameters, regardless of associated Site Collection.

@ParentId: The Work Item Parent identifier of the Work Item. If the parameter is not NULL, then the server MUST only consider for deletion and revert Work Items which have this Work Item Parent identifier. If the parameter is NULL, then the server MUST operate on the Work Items specified by the other parameters, regardless of the value of their Work Item Parent identifier.

@WorkItemType: The Work Item type identifier of the Work Item type. The server MUST only consider for deletion and revert Work Items associated with this Work Item type. MUST NOT be NULL.

@BatchId: The Work Item Batch identifier of the Work Item Batch. If the parameter is not NULL, then the server MUST only consider for deletion and revert Work Items associated with this Work Item Batch. If the parameter is NULL, then the server MUST operate on Work Items specified by the other parameters, regardless of associated Work Item Batch.

@AnyRemaining: Specifies whether the stored procedure reverted any items. The protocol server MUST set this parameter to 1 if it reverted any Work Items. The server MUST set this parameter to 0 if it did not revert any Work Items.

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

Result Sets: MUST NOT return any result sets.

3.1.4.55 proc_UpdateDataViewWhileSaving

The **proc_UpdateDataViewWhileSaving** stored procedure is called to create or update a Data View Web Part or data form Web Part. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_UpdateDataViewWhileSaving(
    @SiteId                 uniqueidentifier,
    @ListId                 uniqueidentifier,
    @ViewId                 uniqueidentifier,
    @DisplayName            nvarchar(255),
    @Type                   tinyint,
    @Flags                  int,
    @PageUrlID              uniqueidentifier,
    @Level                  tinyint
);
```

@SiteId: The Site Collection identifier for the Site Collection. MUST NOT be NULL.

@ListId: The list identifier (as specified in [MS-WSSFO], section 2.2.1.5) of the list. MUST NOT be NULL.

@ViewId: The GUID of the list View. MUST NOT be NULL.

@DisplayName: The Display Name for the Web Part. If this value is NULL the Web Part's Display Name property MUST NOT be updated.

@Type: The Page type (as specified in [MS-WSSFO], section 2.2.3.12) for the Web Part. If this value is NULL the Web Part's Page type property MUST NOT be updated. If this value is Default View the View MUST be made the Default View.

@Flags: The View Flags (as specified in [MS-WSSFO], section 2.2.2.11) for the View. If the VIEWFLAG_MOBILEDEFAULT (0x01000000) bit is set the View MUST be made the Default View for **mobile devices**.

@PageUrlID: The Document identifier (as specified in [MS-WSSFO], section 2.2.1.2) of the Web Part Page. If this parameter is NULL the Document identifier (as specified in [MS-WSSFO], section 2.2.1.2) of the Web Part Page MUST NOT be updated.

@Level: The Publishing Level of the Web Part. MUST NOT be NULL.

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

Value	Description
0	Successful completion.
1	The Data View Web Part was not successfully updated.

Value	Description
212	The specified Site Collection is Locked.
1816	The Quota for the specified Site Collection has been exceeded.

Result Sets: MUST NOT return any result sets.

3.1.4.56 proc_UpdateDocEventReceiver

The proc_UpdateDocEventReceiver stored procedure is called to update the registration of an event receiver for a specified document. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_UpdateDocEventReceiver(
    @DocUrl                 nvarchar(260),
    @Id                     uniqueidentifier,
    @Name                   nvarchar(256),
    @SiteId                 uniqueidentifier,
    @WebId                  uniqueidentifier,
    @ItemId                 int,
    @Type                   int,
    @SequenceNumber         int,
    @Assembly               nvarchar(256),
    @Class                  nvarchar(256),
    @Data                   nvarchar(256),
    @Filter                 nvarchar(256),
    @Credential             int
);
```

@DocUrl: The URL in store-relative form of the document.

@Id: The Event Receiver identifier (as specified in [MS-WSSFO], section 2.2.1.3) of the event receiver. This value MUST NOT be NULL.

@Name: The name of the event receiver. This value MUST NOT be NULL.

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

@WebId: The Site identifier (as specified in [MS-WSSFO], section 2.2.1.10) of the site which contains the document.

@ItemId: Reserved. @ItemId MUST be zero.

@Type: The type of the event receiver. @Type MUST be one of Event Receiver type (as specified in [MS-WSSFO], section 2.2.3.6).

@SequenceNumber: The sequence number (1) of the event receiver. @SequenceNumber MUST be greater than OR equal to zero AND less than OR equal to 65535.

@Assembly: The assembly name of the implementation of the event receiver. This value MUST NOT be NULL.

@Class: The fully qualified class name of the implementation of the event receiver. This value MUST NOT be NULL.

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

@Filter: Reserved. @Filter MUST be NULL.

@Credential: Reserved. @Credential MUST be zero.

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

Value	Description
0	Update succeeded.
3	The document identified by @DocUrl is not found in the site identified by @WebId in the site collection identified by @SiteId.
87	Update failed.

Result Sets: MUST NOT return any result sets.

3.1.4.57 proc_UpdateEventReceiver

The proc_UpdateEventReceiver stored procedure is called to update the registration of a specified event receiver. The T-SQL syntax for the stored procedure is as follows:

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

@Id: The Event Receiver identifier (as specified in [MS-WSSFO], section 2.2.1.3) of the event receiver. This value MUST NOT be NULL.

@Name: The name of the event receiver. This value MUST NOT be NULL.

@SiteId: The Site Collection identifier of the site collection which contains the event host. This value MUST NOT be NULL.

@WebId: The Site identifier (as specified in [MS-WSSFO], section 2.2.1.10) of the site which contains the event host. This value MUST NOT be NULL.

@HostId: The event host identifier of the event host of the event receiver. This value MUST NOT be NULL.

@HostType: The type of the event host of the event receiver. @HostType MUST be one of Event Host type (as specified in [MS-WSSFO], section 2.2.3.5).

@ItemId: Reserved. @ItemId MUST be zero.

@DirName: Reserved. @DirName MUST be NULL.

@LeafName: Reserved. @LeafName MUST be NULL.

@Type: The type of the event receiver. @Type MUST be one of the Event Receiver types specified in [MS-WSSFO], section 2.2.3.6.

@SequenceNumber: The sequence number (1) of the event receiver. @SequenceNumber MUST be greater than OR equal to zero AND less than OR equal to 65535.

@Assembly: The assembly name of the implementation of the event receiver. This value MUST NOT be NULL.

@Class: The fully qualified class name of the implementation of the event receiver. This value MUST NOT be NULL.

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

@Filter: Reserved. @Filter MUST be NULL.

@SourceId: The event receiver source identifier of the event receiver. This is the Feature identifier (as specified in [MS-WSSFO], section 2.2.1.4) of the feature if the event receiver is added via a feature. This is the Content type identifier (as specified in [MS-WSSFO], section 2.2.1.1) of the content type if the event receiver is added via a content type. Otherwise the event receiver source identifier MUST be NULL.

@SourceType: The Event Receiver Source type of the event receiver. @SourceType MUST be one of Event Receiver Source Type values.

@Credential: Reserved. @Credential MUST be 0.

@ContextType: The context type identifier of the event receiver.

@ContextEventType: Reserved. @ContextEventType MUST be NULL.

@ContextId: The context identifier of the event receiver.

@ContextObjectId: The context object identifier for the Event Host of the event receiver.

@ContextCollectionId: The context collection identifier of the event receiver.

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

Value	Description
0	Update succeeded.
87	Update failed because the specified event receiver does not exist in the specified site collection or the site collection does not exist or the type of the event receiver is ContextLookupReceivers Event Receiver Type (as specified in [MS-WSSFO], section 2.2.3.6).

Result Sets: MUST NOT return any result sets.

3.1.4.58 proc_UpdateListFormWhileSaving

The proc_UpdateListFormWhileSaving stored procedure is called to create or update a List Form Web Part. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_UpdateListFormWhileSaving(
    @SiteId                 uniqueidentifier,
    @ListId                 uniqueidentifier,
    @ViewId                 uniqueidentifier,
    @Flags                  int,
    @Type                   tinyint,
    @PageUrlID              uniqueidentifier,
    @Level                  tinyint
);
```

@SiteId: The Site Collection identifier for the Site Collection. MUST NOT be NULL.

@ListId: The list identifier (as specified in [MS-WSSFO], section 2.2.1.5) of the list. MUST NOT be NULL.

@ViewId: The GUID of the list View. MUST NOT be NULL.

@Flags: The View Flags (as specified in [MS-WSSFO], section 2.2.2.11) for the View. If @Flags is NULL the list Form View Flags (as specified in [MS-WSSFO], section 2.2.2.11) are not updated.

@Type: The Page type (as specified in [MS-WSSFO], section 2.2.3.12) for the list Form Web Part. If this value is NULL the list Form Web Part's Page type property MUST NOT be updated.

@PageUrlID: The Document identifier (as specified in [MS-WSSFO], section 2.2.1.2) of the Web Part Page. If this parameter is NULL the Document identifier (as specified in [MS-WSSFO], section 2.2.1.2) of the list Form Web Part MUST not be updated.

@Level: Publishing Level. MUST NOT be NULL.

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

Value	Description
0	Successful completion.
1	The list Form was not successfully Updated.
212	The specified Site Collection is Locked.
1816	The Quota for the specified Site Collection has been exceeded.

Result Sets: MUST NOT return any result sets.

3.1.4.59 proc_UpdateListItemWorkflowInstanceData

The proc_UpdateListItemWorkflowInstanceData stored procedure is called to update a workflow. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_UpdateListItemWorkflowInstanceData(
    @SiteId                  uniqueidentifier,
    @WebId                   uniqueidentifier,
    @ListId                  uniqueidentifier,
    @ItemId                  int,
    @WorkflowInstanceId      uniqueidentifier,
    @InstanceData            image,
    @InstanceDataSize        int,
    @Modifications           ntext,
    @WakeupTime              datetime,
    @InstanceDataVersionId   int,
    @Status1                 int,
    @Status2                 int,
    @Status3                 int,
    @Status4                 int,
    @Status5                 int,
    @Status6                 int,
    @Status7                 int,
    @Status8                 int,
    @Status9                 int,
    @Status10                int,
    @WorkflowCompleted       bit,
    @WorkflowSuspended       bit,
    @WorkflowFaulting        bit,
    @WorkflowTerminated      bit,
    @WorkflowCanceled        bit,
    @UnlockInstance          bit,
    @ProcessingId            uniqueidentifier,
    @InternalState           int OUTPUT
);
```

@SiteId: The Site Collection identifier of the Site Collection which contains the Workflow. The protocol server MUST update the Site Collection Quota to reflect the change in space used by the Workflow.

@WebId: The Site identifier (as specified in [MS-WSSFO], section 2.2.1.10) of the Site which contains the Workflow.

@ListId: The list identifier of the list which contains the list Item the Workflow was created for.

@ItemId: The list Item identifier (as specified in [MS-WSSFO], section 2.2.1.6) of the list Item for which the Workflow was created.

@WorkflowInstanceId: The Workflow identifier of the Workflow. The server MUST NOT update the Workflow if it is a completed workflow. The server MUST set the modification date and time of the workflow to the date and time in UTC when the procedure was called.

@InstanceData: The workflow instance data of the Workflow.

@InstanceDataSize: The size of @InstanceData. If @InstanceData is NULL, @InstanceDataSize MUST contain the value 0.

@Modifications: The Workflow Modifications of the Workflow. If this value is an empty string, or if any of @WorkflowCompleted, @WorkflowCanceled or @WorkflowTerminated contain the value 1, the server MUST NOT update the Workflow Modifications field of the Workflow.

@WakeupTime: The date and time in UTC for the server to wake the Workflow to resume processing. If @WakeupTime is not NULL, @UnlockInstance contains the value 1, and all of @WorkflowCompleted, @WorkflowCanceled and @WorkflowTerminated contain the value 0, the server MUST create a Work Item to wake up the Workflow.

@InstanceDataVersionId: MUST contain the value 0.

@Status1: The Workflow Status1 value for the Workflow. If this value is NOT NULL, the server MUST set the Workflow Status1 field of the Workflow to this value. Otherwise, the server MUST update the Workflow Status1 value as follows:

- If @WorkflowFaulting contains the value 1, the server MUST update the Workflow Status1 value to WFSTAT_FAULTING_RETRY.
- If @WorkflowTerminated contains the value 1 and @WorkflowCompleted contains the value 0, the server MUST update the Workflow Status1 value to WFSTAT_FAULTING.
- If @WorkflowCompleted contains the value 1, the server MUST update the Workflow Status1 value to WFSTAT_COMPLETED.
- If the current Workflow Status1 value is WFSTAT_FAULTING, the server MUST update the Status1 value to WFSTAT_INPROGRESS.
- In other cases, the server MUST NOT update the Workflow Status1 value.

@Status2: This parameter MUST be ignored.

@Status3: This parameter MUST be ignored.

@Status4: This parameter MUST be ignored.

@Status5: This parameter MUST be ignored.

@Status6: This parameter MUST be ignored.

@Status7: This parameter MUST be ignored.

@Status8: This parameter MUST be ignored.

@Status9: This parameter MUST be ignored.

@Status10: This parameter MUST be ignored.

@WorkflowCompleted: Determines whether the Workflow is marked as finished. Once a Workflow is marked as finished, it cannot be marked as not finished. This value MUST NOT be NULL. When @WorkflowCompleted contains the value 1, the server MUST update workflow internal state of the Workflow to add the WFS_COMPLETED flag (0x0004) and remove the WFS_RUNNING (0x0002) and WFS_HASNEWEVENTS New Events (0x0400) flags.

@WorkflowSuspended: Determines whether the Workflow is marked as suspended. This value MUST NOT be NULL. When @WorkflowSuspended contains the value 1, the server MUST update workflow internal state of the Workflow to add the WFS_SUSPENDED (0x0100) flag.

@WorkflowFaulting: Determines whether the Workflow is marked as faulting. This value MUST NOT be NULL. When @WorkflowFaulting contains the value 1, the server MUST update workflow internal state of the Workflow to add the WFS_Faulting flag.

@WorkflowTerminated: Determines whether the Workflow is marked as terminated. Once a Workflow is marked as terminated, it cannot be marked as not terminated. This value MUST NOT be null. When @WorkflowTerminated contains the value 1, the server MUST update workflow internal state of the workflow to add the WFS_Terminated flag.

@WorkflowCanceled: Determines whether the Workflow is marked as canceled. Once a Workflow is marked as canceled, it cannot be marked as not canceled. This value MUST NOT be NULL. When @WorkflowCanceled contains the value 1, the server MUST update workflow internal state of the workflow to add the WFS_Cancelled flag and remove the WFS_Running and WFS_HASNEWEVENTS flags. If any of @WorkflowCompleted, @WorkflowCancelled or @WorkflowTerminated contain the value 1, the server MUST set the workflow instance data for the workflow to null and the instance data size to 0.

@UnlockInstance: Determines whether the Workflow is unlocked. This value MUST NOT be NULL. When @UnlockInstance contains the value 1, or if any of @WorkflowCompleted, @WorkflowCancelled or @WorkflowTerminated contain the value 1, the server MUST update workflow internal state of the workflow to remove the WFS_Locked flag.

@ProcessingId: The workflow process identifier of the workflow process running the Workflow.

@InternalState: The server MUST ignore the input value of this parameter. The server MUST set the output value of this parameter to the workflow internal state of the Workflow after the procedure action is complete.

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

Value	Description
0	Successful completion.
5	Access is denied.
82	Failed to update the workflow.

Result Sets: MUST NOT return any result sets.

3.1.4.60 proc_UpdateListItemWorkflowLock

The proc_UpdateListItemWorkflowLock stored procedure is called to update and lock or unlock a workflow. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_UpdateListItemWorkflowLock(
    @SiteId                 uniqueidentifier,
    @WorkflowInstanceId     uniqueidentifier,
    @Lock                   bit,
    @ProcessingId           uniqueidentifier,
    @EventsNotDelivered     bit = 0
);
```

@SiteId: The Site Collection identifier of the Site Collection containing the Workflow. The server MUST update the Site Collection Quota to reflect the change in space used by the Workflow.

@WorkflowInstanceId: The Workflow identifier of the Workflow to be updated.

@Lock: Determines whether the Workflow will be locked or unlocked. This value MUST NOT be NULL. When set to 1, the server MUST lock the Workflow. The server MUST add the Locked flag (0x0001) to the workflow internal state of the Workflow. When set to 0, the server MUST unlock the Workflow. The server MUST remove the Locked flag (0x0001) from the workflow internal state of the Workflow, and if the workflow internal state contains the Not Started flag (0x0800), the server MUST remove the Not Started flag and add the Running flag (0x0002).

@ProcessingId: The workflow process identifier of the workflow process running the Workflow. If @Lock contains the value 0, the server MUST ignore the value in @ProcessingId.

@EventsNotDelivered: Indicates whether the completed or terminated Workflow has outstanding events. This value MUST NOT be NULL. If @Lock is set to 1, this value MUST be set to 0.

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

Value	Description
0	Successful completion.
5	Error: Access denied.
82	Error: Failed to update or lock the workflow.

Result Sets: MUST NOT return any result sets.

3.1.4.61 proc_UpdateListViewFormWebPartSource

The proc_UpdateListViewFormWebPartSource stored procedure is called to update the Source property of an existing Web Part. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_UpdateListViewFormWebPartSource(
    @SiteId                 uniqueidentifier,
    @WebPartId              uniqueidentifier,
    @Source                 ntext
);
```

@SiteId: The Site Collection identifier of the Site Collection which contains the specified Web Part. MUST NOT be NULL.

@WebPartId: The Web Part identifier (as specified in [MS-WSSFO], section 2.2.1.14) of the Web Part within the Site Collection. MUST NOT be NULL.

@Source: The Web Part properties of the Web Part in either WPV2:WebPart format (as specified in [MS-WPPS], section 2.2.4.2) or HTML format.

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

Value	Description
0	Successful completion.

Value	Description
1	Updating the Web Part Source property failed
13	Web Part for the given @SiteId and @WebPartId does not exist.
212	The specified Site Collection is Locked..
1816	The Quota for the specified Site Collection has been exceeded.

Result Sets: MUST NOT return any result sets.

3.1.4.62 proc_UpdateViewWhileSaving

The proc_UpdateViewWhileSaving stored procedure is called to create or update a list View. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_UpdateViewWhileSaving(
    @SiteId                 uniqueidentifier,
    @ListId                 uniqueidentifier,
    @ViewId                 uniqueidentifier,
    @View                   ntext,
    @DisplayName            nvarchar(255),
    @ContentTypeId          varbinary(512),
    @Type                   tinyint,
    @Flags                  int,
    @BaseViewID             tinyint,
    @PageUrlID              uniqueidentifier,
    @Level                  tinyint
);
```

@SiteId: The Site Collection identifier of the Site Collection. MUST NOT be NULL.

@ListId: The list identifier (as specified in [MS-WSSFO], section 2.2.1.5) of the list. MUST NOT be NULL.

@ViewId: The GUID of the list View. MUST NOT be NULL.

@View: CAML XML of the View.

@DisplayName: The Display Name of the list View. If @Display Name is NULL the Display Name property MUST NOT be updated.

@ContentTypeID: The Content type identifier (as specified in [MS-WSSFO], section 2.2.1.1) of the list Items in the list to be displayed in the list View. If @ContentTypeID is NULL the Content type identifier property MUST NOT be updated.

@Type: The Page type (as specified in [MS-WSSFO], section 2.2.3.12) of the list View. If @Type is NULL the Page type property MUST NOT be updated. If @Type has a value of Default View the View MUST be made the Default View for the list.

@Flags: This field is a bitmask, as specified in View Flags (as specified in [MS-WSSFO], section 2.2.2.11) of the list View. When this property contains null, the View Flags (as specified in [MS-WSSFO], section 2.2.2.11) property MUST NOT be updated. Otherwise, the default list view MUST be set depending on the bit values that are specified in the following table.

Value	Description
VIEWFLAG_MOBILEDEFAULT (0x01000000) bit set	The View for Mobile Devices.
VIEWFLAG_CONTENTTYPEDEFAULT (0x10000000) bit set	If the folders match the Content type identifier (as specified in [MS-WSSFO], section 2.2.1.1) AND the view of the folder is either not selected or not valid THEN Use the list View

@BaseViewID: The base view identifier of the list View. If @BaseViewID is NULL the base view identifier property MUST NOT be updated.

@PageUrlID: The Document identifier (as specified in [MS-WSSFO], section 2.2.1.2) of the Web Part Page. If this parameter is NULL the Document identifier (as specified in [MS-WSSFO], section 2.2.1.2) of the Web Part Page MUST NOT be updated.

@Level: Publishing Level of the list View. MUST NOT be NULL.

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

Value	Description
0	Successful completion.
1	View was not successfully Updated.
212	The specified Site Collection is Locked.
1359	An internal error occurred.
1816	The Quota for the specified Site Collection has been exceeded.

Result Sets: MUST NOT return any result sets.

3.1.4.63 proc_UpdateWebPart

The proc_UpdateWebPart stored procedure is called to update the state of an existing Web Part. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_UpdateWebPart(
    @SiteId                 uniqueidentifier,
    @DirName                nvarchar(256),
    @LeafName               nvarchar(128),
    @Level                  tinyint OUTPUT,
    @bAllUser               bit,
    @SystemID               tSystemID,
    @WebPartID              uniqueidentifier,
    @WebPartTypeID          uniqueidentifier,
    @bCheckLock             bit,
    @IsIncluded             bit,
    @FrameState             tinyint,
    @ZoneID                 nvarchar(64),
    @PartOrder              int,
```

```
    @AllUsersProperties       image,
    @PerUserProperties        image
);
```

@SiteId: The Site Collection identifier of the site collection which contains the Web Part.

@DirName: The Directory Name of the Web Part Page that contains the Web Part.

@LeafName: The Leaf Name of the Web Part Page that contains the Web Part.

@Level: The Publishing Level of the Web Part Page. The value is returned as an output parameter and MUST be the same value passed in or Draft. The value is changed to Draft if the Web Part Page is in a Document Library, @Level is Published, @bCheckLock is 1, @bAllUser is 1, @SystemID references an existing user in the Site Collection, the Web Part Page is Moderated or has minor version control enabled, and creation of a new version of the Web Part Page succeeded.

@bAllUser: Specifies whether to update the Web Part for the Shared View or personal View of the Web Part Page. If this flag is set to 1 the Web Part is updated for the Shared View of the Web Part Page and the changes are available to All Users. If this flag is set to 0 @SystemID is used to update the Web Part for the current users personal View of the Web Part Page and is available only to the current user.

@SystemID: The User identifier (as specified in [MS-WSSFO], section 2.2.1.12) for the current user. If the Web Part Page is moderated or has minor version control enabled then @SystemID is used to track who is modifying the Web Part.

@WebPartID: The Web Part identifier (as specified in [MS-WSSFO], section 2.2.1.14) of the Web Part. MUST NOT be NULL.

@WebPartTypeID: The Web Part type identifier of the Web Part being updated. MUST NOT be NULL.

@bCheckLock: If this flag is set to 1, check if the document is in a state where it can be modified, if it cannot be modified, return specific Return Code values, defined in the following Return Code Values table, that explain why it cannot be modified. If this flag is set to 0, the checks made when this flag is set to 1, are bypassed.

@IsIncluded: The Web Part Is Closed State of the Web Part.

@FrameState: The Web Part Chrome State of the Web Part.

@ZoneID: The name of the Web Part Zone identifier of the Web Part Zone that contains the Web Part.

@PartOrder: The Web Part Zone Index of the Web Part.

@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.

@PerUserProperties: A serialized representation of 0 or more personalizable properties of the Web Part. If this value is NULL then default values will be used for all of the personalizable properties of the Web Part.

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

Value	Description
0	Successful completion.
-2147467259	An error occurred while the stored procedure was running.
2	The Web Part Page cannot be found or @SiteId, @DirName or @LeafName is NULL.
3	The Web Part Page is Moderated or has minor version control enabled, and a new version of the Web Part Page cannot be created.
5	The Web Part being updated is not on the Web Part Page.
12	@bCheckLock is 1, @bAllUser is 0 and @Level is Checked Out.
33	@bCheckLock is 1, @bAllUser is 1, and the Web Part Page is not the Current Version.
87	The Web Part Page is in a Document Library, @Level is Published, @bCheckLock is 1, @bAllUser is 1, @UserId references an existing user in the Site Collection, the Web Part Page is Moderated or has minor version control enabled, and a new Draft version of the Web Part Page cannot be created.
158	@bCheckLock is 1, @bAllUser is 1, @Level is NOT Checked Out and the Web Part Page is required to be Checked Out before it is modified.
212	The Site Collection is Locked.
1816	The Quota for the Site Collection has been exceeded.

Result Sets: MUST NOT return any result sets.

3.1.4.64 proc_UpdateWebPartCache

The proc_UpdateWebPartCache stored procedure is called to write the private data cache of the specified Web Part to the database. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_UpdateWebPartCache(
    @SiteId                 uniqueidentifier,
    @DirName                nvarchar(256),
    @LeafName               nvarchar(128),
    @Level                  tinyint OUTPUT,
    @bAllUser               bit,
    @SystemID               tSystemID,
    @WebPartID              uniqueidentifier,
    @Cache                  image
);
```

@SiteId: The Site Collection identifier of the Site Collection which contains the specified Web Part.

@DirName: The Directory Name of the Web Part Page containing the requested Web Part.

@LeafName: The Leaf Name of the Web Part Page containing the requested Web Part.

@Level: the Publishing Level of the Web Part Page. The value is returned as an output parameter and MUST be the same value as passed into the procedure.

@bAllUser: A bit flag specifying whether to update **Web Part cache** for All Users or just the current user. If this flag is set to "0", proc_UpdateWebPartCache MUST update Web Part Cache just for the current user specified by @SystemID. If this flag is set to "1", proc_UpdateWebPartCache MUST update Web Part Cache for All Users.

@SystemID: The SystemID of the current user.

@WebPartID: The Web Part identifier (as specified in [MS-WSSFO], section 2.2.1.14) of the Web Part.

@Cache: The private data cache of the Web Part.

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

Value	Description
0	Successful completion.
2	The specified Web Part Page cannot be found.
5	The Web Part as specified by @WebPartID exists on a different Web Part Page within the Site Collection.
212	The Site Collection is Locked.
1816	The Quota for the specified Site Collection has been exceeded.
-2147467259	An error occurred while the stored procedure was running.

Result Sets: MUST NOT return any result sets.

3.1.4.65 proc_UpdateWebPartIsIncluded

The proc_UpdateWebPartIsIncluded stored procedure is called to customize or personalize four specific properties of a Web Part: its Web Part Is Closed State, which Web Part Zone it is in, its Web Part Zone Index, and its Web Part Chrome State. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_UpdateWebPartIsIncluded(
    @SiteId                 uniqueidentifier,
    @DirName                nvarchar(256),
    @LeafName               nvarchar(128),
    @Level                  tinyint OUTPUT,
    @bAlluser               bit,
    @UserID                 int,
    @WebPartID              uniqueidentifier,
    @bCheckLock             bit,
    @IsIncluded             bit,
    @ZoneID                 nvarchar(64),
    @PartOrder              int,
    @FrameState             tinyint
);
```

@SiteId: The Site Collection identifier of the site collection which contains the Web Part.

@DirName: The Directory Name of the Web Part Page that contains the Web Part.

@LeafName: The Leaf Name of the Web Part Page that contains the Web Part.

@Level: The Publishing Level of the Web Part Page for the current user. The value is returned as an output parameter and MUST be the same as the input value or Draft. The value is changed to Draft if the Web Part Page is in a Document Library, @Level is Published, @bCheckLock is 1, @bAlluser is 1, @UserID references an existing user in the Site Collection, the Web Part Page is Moderated or has minor version control enabled, and creation of a new version of the Web Part Page succeeded.

@bAlluser: A bit flag specifying whether to update the Web Part for the Shared View or personal View of the Web Part Page. If this flag is set to 1, the Web Part is updated for the Shared View of the Web Part Page and the changes are available to All Users. If this flag is set to 0, the Web Part is updated for the current user's personal View of the Web Part Page.

@UserID: The User identifier (as specified in [MS-WSSFO], section 2.2.1.12) of the current user.

@WebPartID: The Web Part identifier (as specified in [MS-WSSFO], section 2.2.1.14) of the Web Part. This MUST NOT be NULL.

@bCheckLock: If this flag is set to 1, check if the document is in a state where it can be modified, if it cannot be modified, return specific Return Code values, defined in the following Return Code Values table, that explain why it cannot be modified. If this flag is set to 0, the checks made when this flag is set to 1, are bypassed.

@IsIncluded: The Web Part Is Closed State of the Web Part.

@ZoneID: The Web Part Zone identifier of the Web Part Zone in which to put the Web Part.

@PartOrder: The Web Part Zone Index for the Web Part.

@FrameState: The Web Part Chrome State of the Web Part.

Return Code Values: Output parameter @Level MUST return Publishing Level. The stored procedure MUST return an integer listed in the following table:

Value	Description
0	Successful completion.
-2147467259	An error occurred while the stored procedure was running.
2	The Web Part Page cannot be found or @SiteId, @DirName or @LeafName is NULL.
3	The Web Part Page is Moderated or has minor version control enabled, and a new version of the Web Part Page cannot be created.
5	The Web Part is not on the Web Part Page.
12	@bCheckLock is 1, @bAllUser is 0 and @Level is Checked Out.
33	@bCheckLock is 1, @bAllUser is 1, and the Web Part Page is not the Current Version.
87	The Web Part Page is in a Document Library, @Level is Published, @bCheckLock is 1, @bAllUser is 1, @UserId references an existing user in the Site Collection, the Web Part Page is Moderated or has minor version control enabled, and a new Draft version of the Web Part Page cannot be created.
160	The Web Part Page is in a Document Library, @Level is Published, @bCheckLock is 1, @bAllUser is 1, the Web Part Page is Moderated or has minor version control enabled, and

Value	Description
	@UserId is NULL.
212	The Site Collection is Locked.
1816	The Quota for the Site Collection has been exceeded.

Result Sets: MUST NOT return any result sets.

3.1.4.66 proc_UpdateWebPartProps

The proc_UpdateWebPartProps stored procedure is called to update the properties of an existing Web Part. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_UpdateWebPartProps(
    @SiteId                 uniqueidentifier,
    @WebPartID              uniqueidentifier,
    @IsIncluded             bit,
    @FrameState             tinyint,
    @AllUsersproperties     image,
    @PerUserproperties      image,
    @Level                  tinyint OUTPUT
);
```

@SiteId: The Site Collection identifier of the site collection which contains the Web Part.

@WebPartID: The Web Part identifier (as specified in [MS-WSSFO], section 2.2.1.14) of the Web Part. This value MUST NOT be NULL.

@IsIncluded: The Web Part Is Closed State of the Web Part.

@FrameState: The Web Part Chrome State of the Web Part.

@AllUsersProperties: A serialized representation of 0 or more Customizable properties of the Web Part. If this value is NULL then default values will be used for all of the Customizable properties of the Web Part.

@PerUserProperties: A serialized representation of 0 or more personalizable properties of the Web Part. If this value is NULL then default values will be used for all of the personalizable properties of the Web Part.

@Level: The Publishing Level of the Web Part Page containing the Web Part. The value is returned as an output parameter and MUST be the same value passed in.

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

Value	Description
0	Successful completion.
-2147467259	An error occurred while the stored procedure was running.
212	The Site Collection is Locked.
1816	The Quota for the Site Collection has been exceeded.

Result Sets: MUST NOT return any result sets.

3.1.4.67 proc_UpdateWebPartTypeId

The proc_UpdateWebPartTypeId stored procedure is called to update a Web Part's Web Part type identifier property. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_UpdateWebPartTypeId(
    @SiteId                 uniqueidentifier,
    @WebPartID              uniqueidentifier,
    @WebPartTypeId          uniqueidentifier
);
```

@SiteId: The Site Collection identifier of the site collection which contains the Web Part. MUST NOT be NULL.

@WebPartID: The Web Part identifier (as specified in [MS-WSSFO], section 2.2.1.14) of the Web Part within the Site Collection. MUST NOT be NULL.

@WebPartTypeId: New Web Part type identifier of the Web Part. MUST NOT be NULL.

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

Value	Description
0	Successful completion.
-2147467259	An error occurred while the stored procedure was running.

Result Sets: MUST NOT return any result sets.

3.1.4.68 proc_UpdateWebPartWhileSaving

The proc_UpdateWebPartWhileSaving stored procedure is called to add a new Web Part or update an existing Web Part's properties. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_UpdateWebPartWhileSaving(
    @SiteId                 uniqueidentifier,
    @DirName                nvarchar(256),
    @LeafName               nvarchar(128),
    @Level                  tinyint,
    @WebPartID              uniqueidentifier,
    @WebPartTypeID          uniqueidentifier,
    @IsIncluded             bit,
    @FrameState             tinyint,
    @ZoneID                 nvarchar(64),
    @PartOrder              int,
    @ContentTypeID          varbinary(512),
    @Source                 ntext,
    @AllUsersProperties     image,
    @PerUserProperties      image
);
```

@SiteId: The Site Collection identifier of the site collection which contains the requested Web Part.

@DirName: The Directory Name of the Web Part Page containing the Web Part. MUST NOT be NULL.

@LeafName: The Leaf Name of the Web Part Page containing the Web Part. MUST NOT be NULL.

@Level: The Publishing Level of the Web Part. MUST NOT be NULL.

@WebPartID: The Web Part identifier (as specified in [MS-WSSFO], section 2.2.1.14) of the Web Part within the Site Collection. If the @WebPartId matches the Web Part identifier (as specified in [MS-WSSFO], section 2.2.1.14) of an existing Web Part on the different Web Part Page, then a new Web Part identifier (as specified in [MS-WSSFO], section 2.2.1.14) MUST be generated. MUST NOT be NULL.

@WebPartTypeID: The Web Part type identifier of the Web Part. If WebPartTypeID of the existing Web Part is changed and @IsIncluded parameter is NULL and @Level is not equal to LEVEL_CHECKOUT then personalizable properties on the Web Part MUST be deleted. MUST NOT be NULL.

@IsIncluded: The Web Part Is Closed State of the Web Part.

@FrameState: The Web Part Chrome State of the Web Part. MUST NOT be NULL.

@ZoneID: The Web Part Zone identifier of the Web Part Zone of the Web Part.

@PartOrder: The Web Part Zone Index of the Web Part.

@ContentTypeID: The Content type identifier (as specified in [MS-WSSFO], section 2.2.1.1) of the list Items in the list to be displayed in the Web Part.

@Source: The Web Part properties of the Web Part in either WPV2:WebPart format (as specified in [MS-WPPS], section 2.2.4.2) or HTML format.

@AllUsersProperties: A serialized representation of zero or more Customizable properties on the Web Part. If this value is NULL then default values will be used for all of the Customizable properties on the Web Part.

@PerUserProperties: A serialized representation of zero or more personalizable properties on the Web Part. If this value is NULL then default values will be used for all of the personalizable properties on the Web Part.

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

Value	Description
0	Successful completion.
1	An error occurred executing the stored procedure
2	The specified Web Part Page cannot be found.
5	The Web Part being updated is not on the Web Part Page.
33	The specified Web Part Page is not the Current Version.
212	The specified Site Collection is Locked.
1359	An internal error occurred.

Value	Description
1816	The Quota for the specified Site Collection has been exceeded.
-2147467259	An error occurred while the stored procedure was running.

Result Sets: MUST NOT return any result sets.

3.1.4.69 proc_UpdateWorkflowAssociation

The proc_UpdateWorkflowAssociation stored procedure is called to update a workflow association. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_UpdateWorkflowAssociation(
    @Id                     uniqueidentifier,
    @SiteId                 uniqueidentifier,
    @Name                   nvarchar(255),
    @Description            nvarchar(1023),
    @StatusFieldName        nvarchar(64),
    @TaskListId             varbinary(16),
    @HistoryListId          varbinary(16),
    @TaskListTitle          nvarchar(255),
    @HistoryListTitle       nvarchar(255),
    @Configuration          int,
    @AutoCleanupDays        int,
    @PermissionsManual      bigint,
    @InstantiationParams    ntext,
    @Version                int
);
```

@Id: The Workflow association identifier of the Workflow association being updated. The server MUST update the modification date and time of the Workflow association to the date and time in UTC when the stored procedure was called.

@SiteId: The Site Collection identifier of the Site Collection which contains the Workflow association.

@Name: The name of the Workflow association. If this value is NULL, the server MUST NOT update the name field of the Workflow association.

@Description: The description of the Workflow association. If this value is NULL, the server MUST NOT update the description field of the Workflow association.

@StatusFieldName: The name of the Workflow Status field of the Workflow association. If this value is NULL, the server MUST NOT update the Workflow Status field of the Workflow association.

@TaskListId: The list identifier of the Workflow Task list of the Workflow association. If this value is NULL, the server MUST NOT update the Workflow Task list identifier field of the Workflow association.

@HistoryListId: The list identifier of the Workflow History list of the Workflow association. If this value is NULL, the server MUST NOT update the Workflow History list identifier field of the Workflow association.

@TaskListTitle: The title of the Workflow Task list of the Workflow association. If this value is NULL, the server MUST NOT update the Workflow Task list title field of the Workflow association.

@HistoryListTitle: The title of the Workflow History list of the Workflow association. If this value is NULL, the server MUST NOT update the Workflow History list title field of the Workflow association.

@Configuration: The Workflow association Configuration of the Workflow association. This value MUST NOT be NULL.

@AutoCleanupDays: The number of days before Workflows based on the Workflow association are cleaned up. This value MUST contain a positive integer.

@PermissionsManual: The WSS Rights Mask required to manually start any Workflows created from the Workflow association. This value MUST NOT be NULL.

@InstantiationParams: The Workflow association Data of the Workflow association. If this value is NULL, the server MUST NOT update the workflow association data of the Workflow association.

@Version: MUST contain either the value 0 or the current version of the workflow association specified by @Id. The server MUST increment the version of the Workflow association by 1.

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

Value	Description
0	Successful completion.
5	An error occurred.

Result Sets: MUST NOT return any result sets.

3.1.4.70 proc_UpdateWorkItem

The proc_UpdateWorkItem stored procedure is called to modify the properties of a Work Item. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_UpdateWorkItem(
    @WorkItemId              uniqueidentifier,
    @SiteId                  uniqueidentifier,
    @DeliveryDate            datetime,
    @BinaryPayload           image,
    @TextPayload             ntext,
    @ProcessingId            uniqueidentifier,
    @ForceUpdate             bit= 0
);
```

@WorkItemId: The Work Item identifier. The server MUST only update a Work Item if it has the given Work Item identifier. MUST NOT be NULL.

@SiteId: The Site Collection identifier of the Site Collection. The server MUST only update a Work Item if it is associated with this Site Collection. MUST NOT be NULL.

@DeliveryDate: The Work Item Delivery Date. If the parameter is NULL, then the server MUST NOT change the Work Item Delivery Date associated with the Work Item. If the parameter is not NULL, then the server MUST update the Delivery Date of the Work Item to this value. In this case, if the parameter value differs from the previous Delivery Date, then the server MUST:

- Set the Work Item Process identifier associated with the Work Item to NULL,

- Mark the Work Item as not In Progress Work Item, and
- Mark the Work Item as not Throttled Fetch.

@BinaryPayload: The work item binary payload.

@TextPayload: The work item text payload.

@ProcessingId: The Work Item Process identifier of the Work Item Process. If the value of @ForceUpdate is 0, then the server MUST only modify Work Items associated with the Work Item Process indicated by this parameter.

@ForceUpdate: Specifies whether or not the stored procedure MUST update Work Items that do not have the same Work Item Process as the one specified by @ProcessingId. If the value of the parameter is 0, then the server MUST only update the Work Item if the @ProcessingId parameter matches the Work Item Process identifier associated with the Work Item. If the value of the parameter is 1, then the server MUST update the Work Item regardless of the value of the @ProcessingId parameter.

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

Value	Description
0	Successful completion.
5	Error: Access denied.

Result Sets: MUST NOT return any result sets.

3.1.4.71 proc_WorkflowHasVisibleParentItem

The proc_WorkflowHasVisibleParentItem stored procedure is called to determine if the list Item that the Workflow was created for has been deleted. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_WorkflowHasVisibleParentItem(
    @SiteId                 uniqueidentifier,
    @WorkflowInstanceId     uniqueidentifier
);
```

@SiteId: The Site Collection identifier of the Site Collection which contains the Workflow.

@WorkflowInstanceId: The Workflow identifier of the Workflow.

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

Value	Description
0	The list Item that the Workflow was created for has been deleted.
1	The list Item that the Workflow was created for has not been deleted.

Result Sets: MUST NOT return any result sets.

3.1.5 Timer Events

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

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

Refer to section 3.1.1.

3.2.2 Timers

A connection timeout timer is set up on the front-end Web server to govern the total connection time for any requests to the back end database server. The amount of time is governed by a timeout value configured on thefront-end Web server for all back end database server connections.

3.2.3 Initialization

The front-end Web server MUST validate the user making the request before calling the stored procedures. The Site Collection identifier (as specified in [MS-WSSFO], section 2.2.1.8) and the User identifier (as specified in [MS-WSSFO], section 2.2.1.12) for the user making the request are looked up by thefront-end Web server before calling additional stored procedures.

3.2.4 Message Processing Events and Sequencing Rules

The front-end Web server handles each stored procedure with the same processing method of calling the stored procedure and waiting for the 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 Database or Configuration databases, unless explicitly described in this section.

3.2.5 Timer Events

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

3.2.6 Other Local Events

No other local events affect the operation of this protocol.

4 Protocol Examples

This section provides specific example scenarios for manipulating Event receivers, Web Parts, Workflows, and Work Items. These examples describe in detail the process of communication between the front-end Web server and the back end database server.

4.1 Event Receiver

4.1.1 Create an Event Receiver

This example describes the request made and the response returned when a user registers a new event receiver to handle an event for a list in a site.

The user initiates this scenario by registering the new event receiver for the list as specified in the following figure.

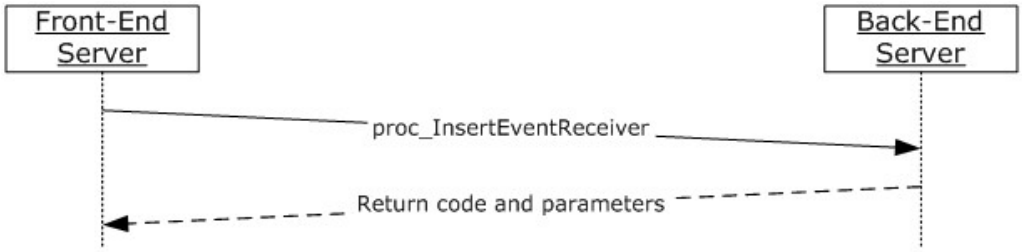

Figure 2: Create an event receiver

1. The user creates a list object that represents the event host list and adds the new event receiver for the desired event to the event host list.
2. The front-end Web server calls proc_InsertEventReceiver to save the event receiver registration to back end database server.
3. The proc_InsertEventReceiver returns a return code.
4. The control returns to the user.

4.1.2 Read Event Receivers

Reading event receivers is part of reading the metadata of the event host. Please refer to [MS-WSSFO2] section 4.4 for an example.

4.1.3 Update an Event Receiver

This scenario is initiated by a user who wants to update an event receiver for a list.

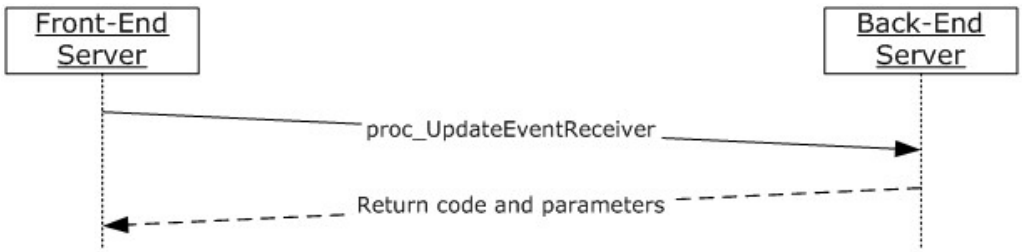

Figure 3: Update an event receiver

1. The user creates a list object that represents the event host list, gets the event receiver to update via the event host list, modifies the properties of the event receiver as desired, then updates the event receiver.

2. The front-end Web server calls proc_UpdateEventReceiver to save the event receiver 's new properties to the back end database server.

3. The proc_UpdateEventReceiver returns a return code.

4. The control returns to the user.

4.1.4 Delete an Event Receiver

This scenario is initiated by a user who wants to delete an event receiver from a list.

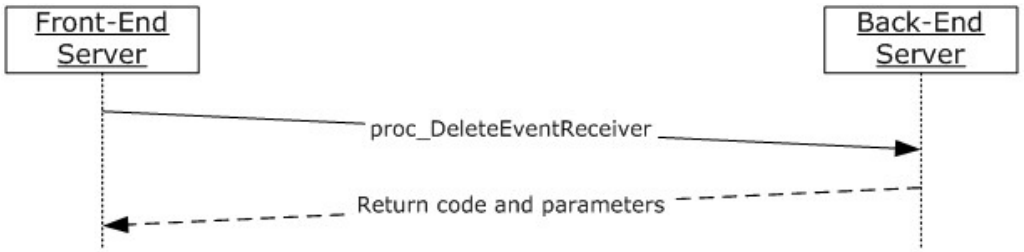

Figure 4: Delete an event receiver

1. The user creates a list object that represents the event host list, gets the event receiver to delete via the event host list, then deletes the event receiver.

2. The front-end Web server calls proc_DeleteEventReceiver to delete the event receiver in the back end database server.

3. The proc_DeleteEventReceiver returns a return code.

4. The control returns to the user.

4.2 Web Part

4.2.1 Add a List View Web Part

This scenario is initiated when a list View Web Part is added to a Web Part Page.

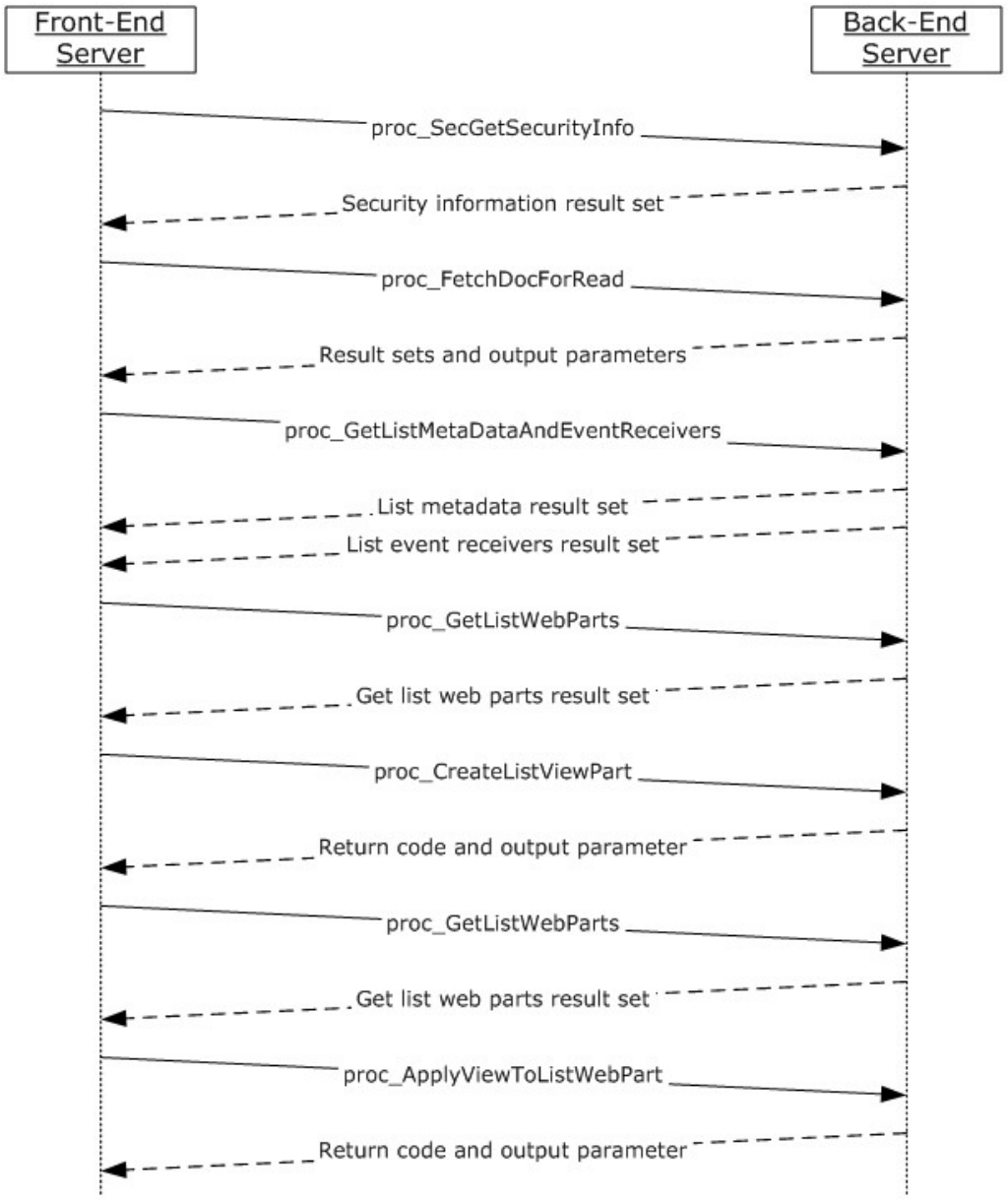

Figure 5: Add a List View Web Part

This example assumes that the list View Web Part to be added is instantiated and refers a valid list and a valid View associated with the list.

The following actions happen:

1. The front-end Web server retrieves security permissions information about the requested Site. It does this by calling the proc_SecGetSecurityInfo stored procedure (as specified in [MS-WSSFO2] section 3.1.5.84).

2. The back-end database server returns the Security Information Result Set, which consists of information about security permissions about the requested Site.

3. The front-end Web server requests information about the Web Part Page to which the list View Web Part needs to be added by calling the [MS-WSSFO2].proc_FetchDocForRead stored procedure (as specified in [MS-WSSFO2] section 3.1.5.20).

4. The Back-End Database Server returns a set of Result Sets as defined in [MS-WSSFO2] section 3.1.5.17.1 through [MS-WSSFO2] section 3.1.5.17.3, and the Publishing Level of the Document as an output parameter.

5. The front-end Web server then fetches properties of the list that the list View Web Part refers to by calling the proc_GetListMetaDataAndEventReceivers stored procedure.

6. The Back-End Database Server returns two Result Sets which include the metadata and Event Receivers for the specified list.

7. The front-end Web server then fetches Views associated with the list by calling the proc_GetListWebParts stored procedure.

8. The Back-End Database Server returns one Result Set which include the list views associated with the list.

9. The front-end Web server sends a request to create a new list View Web Part and a new associated View for the list and place the list View Web Part in the specified Web Part Zone on the specified Web Part Page by calling the proc_CreateListViewPart stored procedure.

10. The Back-End Database Server returns an output code and the Publishing Level as the output parameter.

11. The front-end Web server then re-fetches the Views corresponding to the list by calling the proc_GetListWebParts stored procedure.

12. The Back-End Database Server returns one Result Set which include the list Views corresponding to the list.

13. The front-end Web server then sends a request to copy properties of the View specified in the list View Web Part instance to the newly created View by calling the proc_ApplyViewToListWebPart stored procedure.

14. The Back-End Database Server returns a Return Code status and the View Flags (as specified in [MS-WSSFO], section 2.2.2.11) of the new View for the Web Part as an output parameter.

4.2.2 Add a non-List View Web Part

This scenario is initiated when a Web Part which is NOT a list View Web Part is added to a Web Part Page.

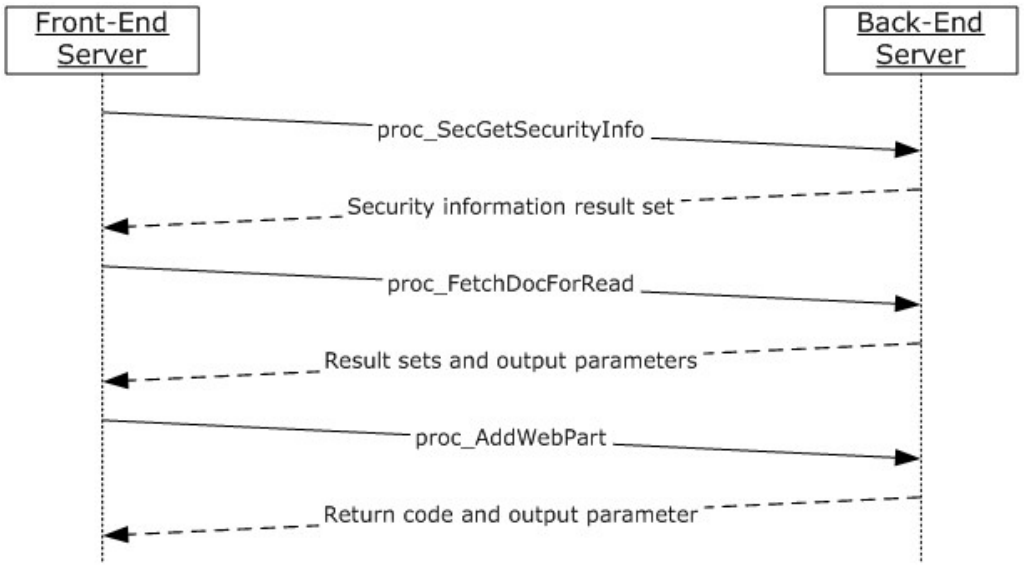

Figure 6: Add a non-List View Web Part

This example assumes the Web Part to be added is instantiated. The following actions happen:

1. The front-end Web server retrieves security permissions information about the requested Site. It does this by calling the [MS-WSSFO2].proc_SecGetSecurityInfo stored procedure (as specified in [MS-WSSFO2] section 3.1.5.84).

2. The Back-End Database Server returns the Security Information Result Set, which consists of information about security permissions about the requested Site.

3. The front-end Web server requests information about the Web Part Page to which the list View Web Part needs to be added by calling the [MS-WSSFO2].proc_FetchDocForRead stored procedure (as specified in [MS-WSSFO2] section 3.1.5.20).

4. The Back-End Database Server returns a set of Result Sets as defined in [MS-WSSFO2] section 3.1.5.17.1 through [MS-WSSFO2] section 3.1.5.17.3, and the Publishing Level of the Document as an output parameter.

5. The front-end Web server sends a request to create a new Web Part and place it in the specified Web Part Zone on the specified Web Part Page by calling the proc_AddWebPart stored procedure.

6. The Back-End Database Server returns an output code and the Publishing Level as the output parameter.

4.2.3 Get All Web Parts on a Web Part Page

This scenario is initiated when a request is made to fetch all the Web Parts on a Web Part Page.

Figure 7: Retrieve all Web Parts on a Web Part Page

The following actions happen:

1. The front-end Web server fetches information about the Web Parts on the Web Part Page by calling the proc_GetAllWebPartsOnPage stored procedure.

2. The Back End Database Server returns Result Sets as specified in proc_GetAllWebPartsOnPage Result Sets.

4.2.4 Delete a Web Part

This scenario is initiated when a Web Part is deleted from a Web Part Page.

Figure 8: Delete a Web Part

This example assumes:

- The Web Part to be deleted is on the specified Web Part Page.
- The Web Part is not personalized.
- The Web Part Page is not contained in a Document Library, or the Document Library containing the Web Part Page has **Required Checkout** set to 0.

The following actions happen:

1. The front-end Web server builds a dynamic T-SQL syntax **query** which requests the particular Web Part to be deleted by calling proc_DeleteWebPart stored procedure. It also queries the return code and the output Publishing Level of the Document from the stored procedure.

2. The Back-End Database Server returns a single Result Set which indicates the Return Code status and output Publishing Level of the Web Part Page.

4.3 Workflow

4.3.1 Create a Workflow for a List Item

This scenario is initiated when a Workflow is added to a list Item.

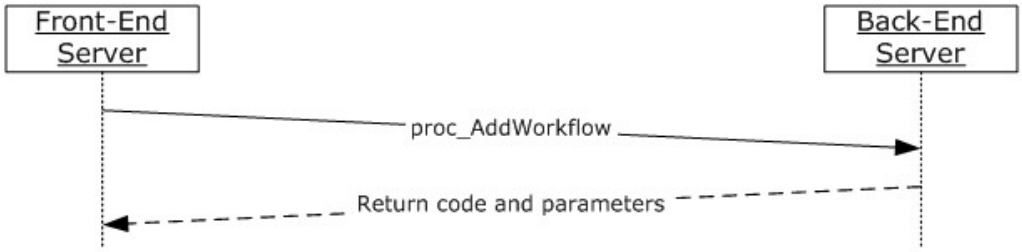

Figure 9: Create a Workflow for a List Item

This example assumes the Workflow to be added refers to a valid list Item and Workflow association associated with the parent list. The following actions happen:

1. The front-end Web server sends a request to create a new Workflow on the specified list Item by calling the proc_AddWorkflow stored procedure.

2. The Back-End Database Server returns a return code specifying the outcome.

4.3.2 Delete a Workflow from a List Item

This scenario is initiated when a Workflow is removed from a list Item.

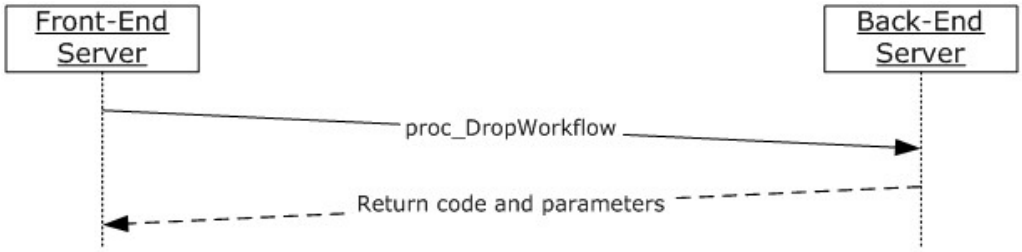

Figure 10: Delete a Workflow from a List Item

This example assumes the Workflow to be removed is instantiated and refers to a valid list Item. The following actions happen:

1. The front-end Web server sends a request to delete an existing Workflow on the specified list Item by calling the proc_DropWorkflow stored procedure.

2. The Back-End Database Server returns a return code specifying the outcome.

4.4 Work Item

4.4.1 Create a Work Item for Bulk Editing Workflow Tasks

This scenario is initiated when a user clicks on a button in the client UI to bulk edit **workflow tasks** with a certain set of values.

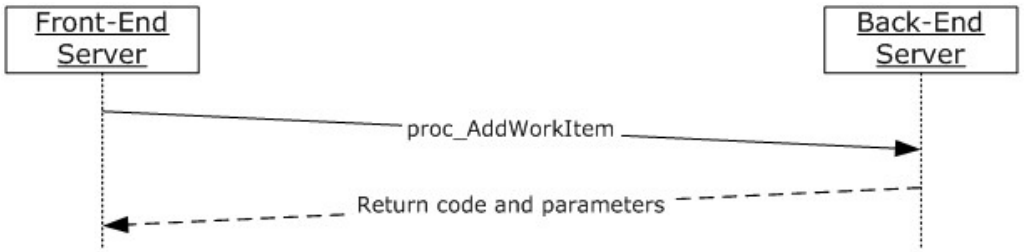

Figure 11: Create a Work Item for Bulk Editing Workflow Tasks

The following actions happen:

1. The front-end Web server requests to create a new Work Item by calling the proc_AddWorkItem stored procedure specifying a work item type identifier representing bulk workflow tasks and a Work Item Delivery Date of the current time to indicate that the work item executes as soon as possible.

2. The Back-End Database Server creates a new Work Item in the Content Database and returns a single Return Code status to indicate whether the Work Item was successfully created.

4.4.2 Retrieve a Set of Runnable Bulk Workflow Task Work Items

This scenario is initiated when a Timer Job runs that executes Work Items of Work Item type Bulk Workflow Task.

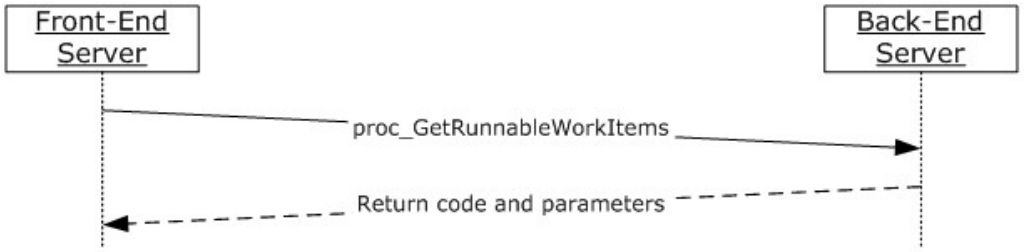

Figure 12: Retrieve a Set of Runnable Bulk Workflow Task Work Items

This example assumes that the Content Database already contains work items whose Delivery dates have passed and are of Work item type bulk workflow task. The following actions happen:

1. The front-end Web server requests the set of Work Items which have delivery dates at or before the current time and are of work item type bulk workflow task by calling the proc_GetRunnableWorkItems stored procedure.

2. The Back-End Database Server returns a set of Work Items and marks them as In Progress Work Items. The Timer Job can then iterate through and run all Work Items in the set.

4.4.3 Delete a Work Item

This scenario is initiated when the Timer Job has finished running a Work Item that have delivery dates that have passed and is about to mark them as finished.

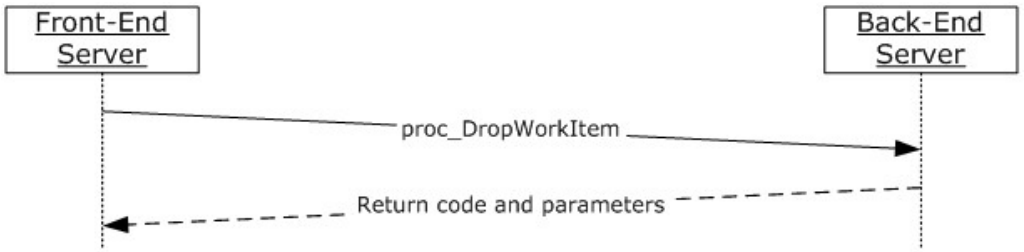

Figure 13: Delete a Work Item

The following actions happen:

1. The front-end Web server requests to mark Work Items as finished by calling the proc_DropWorkItem stored procedure.

2. The Back-End Database Server deletes the work item and returns a single Return Code status to indicate execution completion.

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 calling the stored procedure.

5.2 Index of Security Parameters

None.

6 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- Microsoft® SQL Server® 2005
- Microsoft® SQL Server® 2008
- Microsoft® SQL Server® 2008 R2
- Windows® SharePoint® Services 3.0

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

<1> Section 2.2.6.4: Section 2.2.5.4: SharePoint Products and Technologies MAY use 1 as an arbitrary placeholder when there is no list item associated with the work item.

<2> Section 3.1.4.6: Section 3.1.4.6: SharePoint Products and Technologies MAY use 1 as an arbitrary placeholder when there is no list item associated with the work item.

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

T

U

V

W

X