

[MS-WSSPROG2]: Windows SharePoint Services: Content Database Programmability Extensions Communications Version 2 Protocol Specification

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

Open Specifications are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

Revision Summary

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

Table of Contents

1	Introduction	8
1.1	Glossary	8
1.2	References.....	11
1.2.1	Normative References.....	11
1.2.2	Informative References	11
1.3	Protocol Overview (Synopsis)	11
1.3.1	Event Operations.....	12
1.3.2	Web Part Operations.....	12
1.3.3	Workflow Operations	12
1.3.4	Work Item Operations.....	12
1.4	Relationship to Other Protocols.....	12
1.5	Prerequisites/Preconditions	13
1.6	Applicability Statement.....	13
1.7	Versioning and Capability Negotiation.....	13
1.8	Vendor-Extensible Fields.....	13
1.9	Standards Assignments	15
2	Messages.....	16
2.1	Transport.....	16
2.2	Common Data Types	16
2.2.1	Simple Data Types and Enumerations	16
2.2.2	Simple Data Types	16
2.2.2.1	Context Collection Identifier	16
2.2.2.2	Context Identifier	16
2.2.2.3	Context Object Identifier	16
2.2.2.4	Context Type Identifier.....	16
2.2.2.5	Event Receiver Source Identifier.....	16
2.2.2.6	List Item Version	16
2.2.2.7	Workflow Template Identifier	16
2.2.3	Bit Fields and Flag Structures.....	16
2.2.3.1	Event Receiver Source Type.....	16
2.2.3.2	Workflow Association Configuration.....	17
2.2.3.3	Workflow Internal State.....	18
2.2.3.4	Workflow Status1	18
2.2.4	Enumerations	19
2.2.4.1	Sandboxed Solution Status	19
2.2.5	Binary Structures	19
2.2.6	Result Sets.....	19
2.2.6.1	List Item Workflows Result Set.....	19
2.2.6.2	Solution Resource Usage Result Set	21
2.2.6.3	Solution Resource Usage Processing Result Set	21
2.2.6.4	Web Parts Result Set	22
2.2.6.5	Workflow Associations Result Set.....	23
2.2.6.6	Work Items Result Set	25
2.2.7	Tables and Views	26
2.2.8	XML Structures	26
2.2.8.1	Namespaces	26
2.2.8.2	Simple Types	26
2.2.8.3	Complex Types.....	26
2.2.8.4	Elements	26

2.2.8.4.1	Workflow Modifications	26
2.2.8.5	Attributes	27
2.2.8.6	Groups	27
2.2.8.7	Attribute Groups	27
3	Protocol Details	28
3.1	Server Details	28
3.1.1	Abstract Data Model	28
3.1.1.1	Web Parts	28
3.1.1.1.1	Customizable and Personalizable Properties	28
3.1.1.1.2	Adding and Modifying a Web Part for All Users (Customization)	28
3.1.1.1.3	Adding a Web Part for All Users then modifying it uniquely for a particular User (Personalization)	28
3.1.1.1.4	Adding a Web Part just for a particular User (Personal Web Part)	29
3.1.1.1.5	Versioning Web Parts Pages	29
3.1.1.1.6	Changing a Web Part Type Identifier	29
3.1.1.1.7	Web Part Caching	30
3.1.1.2	Workflow	30
3.1.1.2.1	Workflow Concepts	30
3.1.1.2.2	Workflow Reusability	30
3.1.1.3	Work Items	30
3.1.1.4	Event Receivers	30
3.1.1.4.1	Event Receiver Concepts	30
3.1.1.4.2	Registering Event Receivers	30
3.1.1.4.3	Scopes of Event Receivers	30
3.1.1.4.4	Sequences of Event Receivers	31
3.1.1.5	Quota Management	31
3.1.1.6	Sandboxed Solution Resource Usage Monitoring	31
3.1.2	Timers	31
3.1.3	Initialization	31
3.1.4	Message Processing Events and Sequencing Rules	31
3.1.4.1	proc_AddNonListViewFormPersonalization	31
3.1.4.2	proc_AddNonListViewFormWebPartForUrl	33
3.1.4.3	proc_AddSolution	35
3.1.4.4	proc_AddWebPart	36
3.1.4.5	proc_AddWorkflow	38
3.1.4.6	proc_AddWorkflowAssociation	40
3.1.4.7	proc_AddWorkItem	41
3.1.4.8	proc_ApplyViewToListWebPart	43
3.1.4.9	proc_AutoCleanupWorkflows	45
3.1.4.10	proc_AutoDropWorkflows	45
3.1.4.11	proc_CancelDeclarativeWorkflows	47
3.1.4.12	proc_CancelWorkflow	47
3.1.4.13	proc_CommitUpdatedZoneIds	47
3.1.4.14	proc_CompleteInProgressWorkItems	48
3.1.4.15	proc_CopyDefaultViewWebParts	49
3.1.4.16	proc_CountWorkflowAssociations	50
3.1.4.16.1	Count Workflow Associations Result Set	50
3.1.4.17	proc_CountWorkflows	50
3.1.4.17.1	Count Workflows Result Set	51
3.1.4.18	proc_CountWorkflowsBatch	51
3.1.4.18.1	Workflows Batch Result Set	52
3.1.4.19	proc_CreateListViewPart	52

3.1.4.20	proc_DeleteDocEventReceiver	54
3.1.4.21	proc_DeleteEventReceiver.....	55
3.1.4.22	proc_DeleteEventReceiversBySourceId	57
3.1.4.23	proc_DeleteInProgressWorkItems	58
3.1.4.24	proc_DeleteSmartPagePersonalization	59
3.1.4.25	proc_DeleteWebPart.....	59
3.1.4.26	proc_DeleteWebPartPersonalization.....	61
3.1.4.27	proc_DeleteWebPartWhileSaving	62
3.1.4.28	proc_DeleteZoneWebPartsWhileSaving	62
3.1.4.29	proc_DisableAssociationsForTemplate.....	63
3.1.4.30	proc_DropWorkflow	64
3.1.4.31	proc_DropWorkflowAssociation	64
3.1.4.32	proc_DropWorkItem.....	65
3.1.4.33	proc_EnableDeclarativeWorkflowAssociations	65
3.1.4.34	proc_EnumerateWebPartsForList.....	66
3.1.4.34.1	Web Parts Result Set	66
3.1.4.35	proc_EnumerateWebPartsForWeb	66
3.1.4.35.1	Web Parts Result Set	67
3.1.4.36	proc_EnumResourceWarningSites	68
3.1.4.36.1	Resource Warning Site Collections Result Set	68
3.1.4.37	proc_FailOverInProgressWorkItems.....	69
3.1.4.38	proc_GetAllResourceUsageForSiteToday	69
3.1.4.38.1	Site Collection Daily Resource Usage Result Set	69
3.1.4.39	proc_GetAllWebPartsOnPage	70
3.1.4.39.1	Web Parts Metadata, Non-Personalized Result Set	71
3.1.4.39.2	Web Parts Metadata, Personalized Result Set	71
3.1.4.39.3	List Metadata, Result Set.....	71
3.1.4.39.4	List Event Receivers, Result Set	71
3.1.4.39.5	List Security Information, Result Set.....	71
3.1.4.40	proc_GetAverageDailyResourceUsageForSite	71
3.1.4.40.1	Site Collection Average Daily Resource Usage Result Set	72
3.1.4.41	proc_GetContextCollectionEventReceivers	72
3.1.4.41.1	Event Receivers Result Set	72
3.1.4.42	proc_GetContextObjectEventReceivers	73
3.1.4.42.1	Event Receivers Result Set	74
3.1.4.43	proc_GetDocEventReceivers.....	74
3.1.4.43.1	Event Receivers Result Set	74
3.1.4.44	proc_GetListItemWorkflows.....	74
3.1.4.44.1	List Item Workflows Result Set.....	76
3.1.4.45	proc_GetListItemWorkflowWithInstanceDataAndLock	76
3.1.4.46	proc_GetListWebParts	77
3.1.4.46.1	List Web Parts Result Set	77
3.1.4.47	proc_GetNextWebPartOrder	79
3.1.4.48	proc_GetRecycleBinItemEventReceivers.....	79
3.1.4.48.1	Recycle Bin Item Result Set	80
3.1.4.48.2	List Event Receivers Result Set	81
3.1.4.48.3	Site Event Receivers Result Set	81
3.1.4.49	proc_GetRunnableWorkItems	81
3.1.4.49.1	Work Items Result Set	82
3.1.4.50	proc_GetRunningWorkBatchCount.....	82
3.1.4.51	proc_GetSiteResourceUsage.....	83
3.1.4.51.1	Site Collection Resource Usage Result Set	83
3.1.4.52	proc_GetSiteSolutionResourceUsage	83

3.1.4.52.1	Site Solution Resource Usage Result Set	84
3.1.4.53	proc_GetSolutionInfo	84
3.1.4.53.1	Solution Hash Information Result Set.....	84
3.1.4.54	proc_GetSolutionResourceQuota	85
3.1.4.54.1	Solution Resource Quota Result Set	85
3.1.4.55	proc_GetSolutionResourceUsage	85
3.1.4.55.1	Solution Resource Usage Result Set	86
3.1.4.56	proc_GetSolutionResourceUsageDailyOrdinal	86
3.1.4.56.1	Solution Resource Usage Daily Ordinal Result Set.....	86
3.1.4.57	proc_GetSolutionsData	86
3.1.4.57.1	Solution Data Result Set.....	87
3.1.4.58	proc_GetWFTemplatesLastModifiedForWeb.....	87
3.1.4.59	proc_GetWorkflowAssociations	88
3.1.4.59.1	Workflow Associations Result Set	89
3.1.4.60	proc_GetWorkflowDataForItem	89
3.1.4.60.1	Workflow Associations Result Set	90
3.1.4.60.2	List Item Workflows Result Set.....	90
3.1.4.61	proc_GetWorkItems	90
3.1.4.61.1	Single Work Item Result Set	91
3.1.4.61.2	Multiple Work Items Result Set	91
3.1.4.62	proc_InsertContextEventReceiver	91
3.1.4.63	proc_InsertDocEventReceiver	93
3.1.4.64	proc_InsertEventReceiver	95
3.1.4.65	proc_LogSolutionResourceUsage20	97
3.1.4.66	proc_LogSolutionResourceUsageDaily20	99
3.1.4.67	proc_LogSolutionResourceUsageWindowed20	102
3.1.4.68	proc_ProcessSolutionResourceUsageLogData.....	105
3.1.4.68.1	Solution Resource Usage Log Processing Result Set	105
3.1.4.69	proc_ProcessSolutionResourceUsageWindowedData	105
3.1.4.69.1	Windowed Solution Resource Usage Processing Result Set.....	106
3.1.4.70	proc_ProvisionWebPart	106
3.1.4.71	proc_RemoveSolution.....	107
3.1.4.72	proc_ResetSiteResourceUsageWarnings.....	107
3.1.4.73	proc_RestoreWebPartForDoc	108
3.1.4.74	proc_RevertInProgressWorkItem	108
3.1.4.75	proc_RevertInProgressWorkItems	109
3.1.4.76	proc_SetEventReceiverToSynchronous	110
3.1.4.77	proc_TruncateResourceUsageDaily	110
3.1.4.78	proc_TruncateResourceUsageLog	111
3.1.4.79	proc_TruncateResourceUsageWindowed	111
3.1.4.80	proc_UpdateDataViewWhileSaving	111
3.1.4.81	proc_UpdateDocEventReceiver	112
3.1.4.82	proc_UpdateEventReceiver.....	114
3.1.4.83	proc_UpdateListFormWhileSaving	116
3.1.4.84	proc_UpdateListItemWorkflowInstanceData	117
3.1.4.85	proc_UpdateListItemWorkflowLock.....	120
3.1.4.86	proc_UpdateListViewFormWebPartSource	121
3.1.4.87	proc_UpdateListViewToDataViewForSite.....	121
3.1.4.88	proc_UpdateListViewToDataViewForWeb	122
3.1.4.89	proc_UpdateSiteResourceUsage	122
3.1.4.90	proc_UpdateSolution	123
3.1.4.91	proc_UpdateSolutionResourceUsage.....	124
3.1.4.92	proc_UpdateViewWhileSaving.....	124

3.1.4.93	proc_UpdateWebPart.....	126
3.1.4.94	proc_UpdateWebPartCache	128
3.1.4.95	proc_UpdateWebPartIsIncluded	129
3.1.4.96	proc_UpdateWebPartProps	130
3.1.4.97	proc_UpdateWebPartTypeId	132
3.1.4.98	proc_UpdateWebPartWhileSaving.....	132
3.1.4.99	proc_UpdateWorkflowAssociation.....	134
3.1.4.100	proc_UpdateWorkItem.....	136
3.1.4.101	proc_WorkflowHasVisibleParentItem.....	137
3.1.5	Timer Events	137
3.1.6	Other Local Events	137
3.2	Client Details.....	138
3.2.1	Abstract Data Model	138
3.2.2	Timers	138
3.2.3	Initialization	138
3.2.4	Message Processing Events and Sequencing Rules.....	138
3.2.5	Timer Events	138
3.2.6	Other Local Events	138
4	Protocol Examples.....	139
4.1	Event Receiver	139
4.1.1	Create an Event Receiver	139
4.1.2	Read Event Receivers	139
4.1.3	Update an Event Receiver	139
4.1.4	Delete an Event Receiver	140
4.2	Web Part	140
4.2.1	Add a List View Web Part	140
4.2.2	Add a non-List View Web Part	142
4.2.3	Get All Web Parts on a Web Part Page.....	143
4.2.4	Delete a Web Part	144
4.3	Workflow	144
4.3.1	Create a Workflow for a List Item	144
4.3.2	Delete a Workflow from a List Item	145
4.4	Work Item	145
4.4.1	Create a Work Item for Bulk Editing Workflow Tasks.....	145
4.4.2	Retrieve a Set of Runnable Bulk Workflow Task Work Items	146
4.4.3	Delete a Work Item	146
5	Security.....	148
5.1	Security Considerations for Implementers.....	148
5.2	Index of Security Parameters	148
6	Appendix A: Product Behavior.....	149
7	Change Tracking.....	150
8	Index	151

1 Introduction

This document specifies the Windows SharePoint Services: Content Database Programmability Extensions Communications Protocol. This protocol specifies the communication sequences used by a protocol client to perform data query and update operations on a protocol server in relation to Web Part, event receiver, workflow, and work item data.

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 [\[MS-OFCGLOS\]](#):

after event receiver
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
daily solution resource usage log
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
empty GUID
event
event host
event receiver

event receiver source
exponential backoff
feature
feature definition
folder
front-end Web server
fully qualified class name
hash
immediate solution resource usage log
item identifier
leaf name
list
List Form Web Part
list identifier
list item
list item identifier
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
request identifier
result set
return code
sandboxed solution
sequence number
shared view
site
site collection
site collection identifier
site solution
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
windowed solution resource usage log
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:

before event receiver: A code routine that processes a synchronous event whose handler runs completely before the action that raised the event is finalized.

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

resource measure: A mechanism for measuring and aggregating the use of a specific resource, such as memory or CPU, for a specified number of time intervals.

resource usage measurement: A floating-point number that indicates the use of a resource by a sandboxed solution for a specific time interval.

resource usage quotas: A built-in or user-defined limit on the use of resources, such as memory and CPU, by processes related to code that has been deployed to a server.

resource usage value: A floating-point number that indicates the use of a resource by a sandboxed solution for a specified number of time intervals.

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

1.2 References

1.2.1 Normative References

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

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

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

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

1.3 Protocol Overview (Synopsis)

This protocol specifies the communication between the **front-end Web server** and the **back-end database server**. This communication satisfies requests associated with **events**, **Web Parts**, **workflows**, and **work items**. This client/server protocol uses the Tabular Data Stream Protocol as defined in [\[MS-TDS\]](#) as its transport between the front-end Web server, 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 returns it 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 returns it 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 returns it 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 returns it 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:

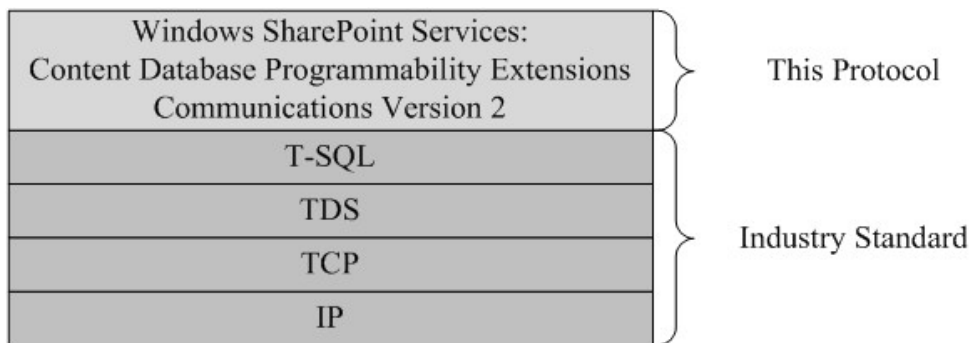


Figure 1: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

The operations described by the protocol operate between a front-end Web server 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 front-end Web server has appropriate permissions to call the stored procedures 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:

sandboxed solution hash data – A binary structure that contains a hash of a sandboxed solution or the sandboxed solution validators for the sandboxed solution. This binary can be passed into or retrieved by the following stored procedures and result sets:

- [proc_AddSolution](#)
- [proc_RemoveSolution](#)
- [proc_UpdateSolution](#)
- [Solution Hash Information Result Set](#)
- [Solution Data Result Set](#)

Web Part customizable and personalizable properties – A binary structure that contains zero or more serialized customizable or personalizable properties for a Web Part. This binary can be passed into or retrieved by the following stored procedures and result sets:

- [proc_AddNonListViewFormWebPartForUrl](#)
- [proc_AddWebPart](#)
- [proc_CreateListViewPart](#)
- [proc_UpdateWebPart](#)
- [proc_UpdateWebPartProps](#)
- [proc_UpdateWebPartWhileSaving](#)
- [List Web Parts Result Set](#)
- [Web Parts Result Set](#)

workflow instance data – A binary payload that contains the state of a workflow. This binary can be passed into or retrieved by the following stored procedures and result sets:

- [proc_UpdateListItemWorkflowInstanceData](#)
- [List Item Workflows Result Set](#)

work item binary payload – A binary payload 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 stored procedures and result sets:

- [proc_AddWorkItem](#)
- [proc_UpdateWorkItem](#)
- [Work Items Result Set](#)

work item text payload – A text payload 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 stored procedures and result sets:

- `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 stored procedures 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 stored procedures and result sets:

- [proc_AddWorkflowAssociation](#)
- [proc_UpdateWorkflowAssociation](#)
- `Workflow Associations Result Set`

Web Part cache data – A binary payload that contains cached information about a web part. This data can be passed into or retrieved by the following stored procedures and result sets:

- `proc_AddNonListViewFormWebPartForUrl`
- [proc_UpdateWebPartCache](#)
- List Web Parts Result Set
- Web Parts 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

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

2.2.2.5 Event Receiver Source Identifier

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**.

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 mask are bitwise combinations of the values listed 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 synchronously because the number of running workflows exceeds the configured limit, it will be deferred for later processing. If this flag is not set, the workflow will always be started synchronously, regardless of the number of running workflows.

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 mask are bitwise combinations of the values listed in the following table.

Value	Name	Meaning
0x00000001	WFS_LOCKED	A front-end Web server has locked the workflow for processing. 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 an error and will be terminated.
0x00000080	WFS_TERMINATED	The workflow was terminated by an unrecoverable error 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

2.2.4.1 Sandboxed Solution Status

The status of the sandboxed solution. The possible values are listed in the following table:

Value	Description
0	The sandboxed solution has not been activated.
1	The sandboxed solution has been activated.
2	The sandboxed solution has been temporarily disabled because it exceeded its resource quota.

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	nvarchar(max),
InstanceData	varbinary(max),
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 [2.2.1.11](#), 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: The [Workflow 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 Solution Resource Usage Result Set

The Solution Resource Usage Result Set contains **resource usage values** for a sandboxed solution. The T-SQL syntax for the result set is as follows:

SolutionId	uniqueidentifier NOT NULL,
ResourceId	uniqueidentifier NOT NULL,
RelativeDaysAgo	int NOT NULL,
SampleCount	int NOT NULL,
ResourceUsage	float NOT NULL;

SolutionId: The identifier of the sandboxed solution.

ResourceId: The identifier of a monitored **resource measure**.

RelativeDaysAgo: The number of monitoring intervals since this resource usage value was collected.

SampleCount: The number of sample points taken for this resource usage value.

ResourceUsage: The resource usage value for the given sandboxed solution, monitored resource measure, and monitoring interval for resource usage.

2.2.6.3 Solution Resource Usage Processing Result Set

The Solution Resource Usage Processing Result Set is used when aggregating **resource usage measurements**. The T-SQL syntax for the result set is as follows:

SiteId	uniqueidentifier NOT NULL,
SolutionId	uniqueidentifier NOT NULL,
ResourceId	uniqueidentifier NOT NULL,
StartTime	datetime NOT NULL,
EndTime	datetime NOT NULL,
SampleCount	int NOT NULL,
ResourceUsage	numeric NOT NULL,
Id	bigint NOT NULL;

SiteId: The site collection identifier of the site collection that contains the sandboxed solution for this resource usage measurement.

SolutionId: The identifier of the sandboxed solution for this resource usage measurement.

ResourceId: The identifier monitored resource measure for this resource usage measurement.

StartTime: The beginning of the time interval, in the local time zone of the front-end Web server, corresponding to this resource usage measurement.

EndTime: The end of the time interval, in the local time zone of the front-end Web server, corresponding to this resource usage measurement.

SampleCount: The number of sample points taken for this resource usage measurement.

ResourceUsage: The resource usage measurement for the given sandboxed solution and monitored resource measure.

Id: The identifier for this resource usage measurement.

2.2.6.4 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_Assembly	nvarchar(255),
tp_Class	nvarchar(255),
tp_SolutionId	uniqueidentifier,
tp_AllUsersProperties	varbinary(max),
tp_PerUserProperties	varbinary(max),
tp_Cache	varbinary(max),
tp_Source	nvarchar(max);

tp_ID: The Web Part Identifier, as specified in [\[MS-WSSFO\]](#), section [2.2.1.15](#). 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 [\[MS-WSSFO\]](#), 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: 1 if the Web Part is included the Web Part Page; 0 if the Web Part is not included. 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_Assembly: The fully qualified name of the assembly that implements the web part.

Tp_Class: The name of the .NET class that implements the Web Part.

Tp_SolutionId: The identifier of the sandboxed solution or site solution that installed the Web Part.

Tp_AllUsersProperties: A binary payload containing 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 binary payload containing 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 WPV2:WebPart format (as specified in [\[MS-WPPS\]](#), section [2.2.3.2](#)), WPV3:WebPart format (as specified in [\[MS-WPPS\]](#), section [2.2.3.3](#)) or WebParts format (as specified in [\[MS-WPPS\]](#), section [2.2.3.1](#)). The protocol client can determine which format is used by comparing the value against the schemas for the formats. The value will be NULL if the properties are compressed and stored in Tp_AllUserProperties and Tp_PerUserProperties.

2.2.6.5 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	nvarchar(max),
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 [2.2.1.11](#)) 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 [\[MS-WSSFO\]](#), section [2.2.2.13](#)) 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 deleted by the back-end database server.

InstantiationParams: The workflow association data of the workflow.

Configuration: The [Workflow association Configuration](#) value for the workflow association.

2.2.6.6 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	varbinary(max),
TextPayload	nvarchar(max),
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: The work item **binary payload**.

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

Prefix	Namespace URI	Reference
s	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1] , [XMLSCHEMA2]

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:complexType>
```

```

    <s:sequence>
      <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="TemplateId" type="s:string " minOccurs="0" maxOccurs="1"/>
      <s:element name="Data" type="s:string" minOccurs="1" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</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: A string containing a GUID in which any alphabetic characters MUST be in upper case. The GUID identifies the parent **Mod** element.

Mod.TemplateId: A string containing a GUID in which any alphabetic characters MUST be in upper case. This GUID is shared among **Mod** elements in **Mods** that share some vendor-extensible characteristic.

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 [2.6](#), 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 of these event receivers. The event receiver with the smallest sequence number 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 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.

3.1.1.6 Sandboxed Solution Resource Usage Monitoring

As part of enforcing solution **resource usage quotas**, the protocol server implements a series of logs: the **immediate solution resource usage log**, the **windowed solution resource usage log**, and the **daily solution resource usage log**. The protocol client uses these logs to track the resource usage values for sandboxed solutions. To enable efficient management of these logs, the protocol server maintains an ordinal used for monitoring resource usage over a given interval that is incremented after every monitoring interval for resource usage.

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     varbinary(max),
    @RequestGuid       uniqueidentifier = NULL OUTPUT
);

```

@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 [2.2.1.15](#)) of the Web Part to be personalized. This MUST NOT be NULL.

@UserId: The User Identifier (as specified in [\[MS-WSSFO\]](#), section [2.2.1.13](#)) 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.

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

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,
    @Assembly               nvarchar(255),
    @Class                  nvarchar(255),
    @SolutionId             uniqueidentifier,
    @AllUsersProperties      varbinary(max),
    @PerUserProperties       varbinary(max),
    @Cache                  varbinary(max),
    @Source                 nvarchar(max),
    @UserId                 int = NULL,
    @Level                  tinyint = 1,
    @BaseViewId             tinyint = NULL,
    @bHasFGP                bit = NULL,
    @bDeleteUsersOtherWebParts bit = 0,
    @bRetainObjectIdentity  bit = 0,
    @View                   varbinary(max) = NULL,
    @RequestGuid            uniqueidentifier = NULL OUTPUT
);
```

@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 [2.2.1.15](#)) 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 [2.2.3.14](#)) 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.

@Assembly: The fully qualified name of the assembly that implements the web part.

@Class: The name of the .NET class that implements the web part.

@SolutionId: The identifier of the sandboxed solution that implements the web part.

@AllUsersProperties: A binary payload containing zero or more customizable properties on the Web Part.

@PerUserProperties: A binary payload containing 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 [2.2.3.2](#)) or HTML format.

@UserId: The User Identifier (as specified in [MS-WSSFO], section [2.2.1.13](#)) 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.

@bRetainObjectIdentity: If this parameter is set to 1 and a Web Part identified by the @SiteId, @WebPartId and @Level parameters exists in a Web Part Page different from the Web Part Page identified by the @SiteId, @DocDirName and @DocLeafName parameters, the server MUST update the existing Web Part and move it to the Web Part Page rather than adding a new web part.

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

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

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_AddSolution**

The `proc_AddSolution` stored procedure is called to add a sandboxed solution to a site collection. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ AddSolution (
    @Name                nvarchar(128),
    @SiteId              uniqueidentifier,
    @SolutionId          uniqueidentifier,
    @Hash                nvarchar(50),
    @ValidatorsHash      char(64),
    @SolutionGalleryItemId int,
    @Status              smallint,
    @HasAssemblies       tinyint,
    @Definitions         varbinary(max)
);
```

@Name: The name of the sandboxed solution.

@SiteId: The Site Collection identifier of the Site Collection on which the sandboxed solution is to be added. This value MUST NOT be NULL.

@SolutionId: The identifier of the sandboxed solution. This value MUST NOT be NULL.

@Hash: The implementation-specific hash of the content of the sandboxed solution. This value MUST NOT be NULL.

@ValidatorsHash: An implementation-specific **hash** that uniquely identifies the validation programs that were run on the sandboxed solution. This value is determined by the implementation of the protocol server. This value MUST NOT be NULL.

@SolutionGalleryItemId: The **list item identifier** of the List Item in Solution Gallery List that contains this solution. This value MUST NOT be NULL.

@Status: This value MUST be 1.

@HasAssemblies: MUST be 1 if the sandboxed solution contains assemblies. MUST be 0 otherwise.

@Definitions: The implementation-specific serialization of the **feature definitions** for this sandboxed solution. This value is determined by the implementation of the protocol server.

Return values: An integer which MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.4.4 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,
    @Assembly               nvarchar(255),
    @Class                  nvarchar(255),
    @SolutionId             uniqueidentifier,
    @TheListID              uniqueidentifier,
    @bCheckLock             bit,
    @IsIncluded             bit,
    @FrameState             tinyint,
    @ZoneID                 nvarchar(64),
    @PartOrder              int,
    @TheFlags               int,
    @TheType                tinyint,
    @TheBaseViewID          tinyint,
    @AllUsersProperties      varbinary(max),
    @PerUserProperties       varbinary(max),
    @RequestGuid            uniqueidentifier = NULL OUTPUT
);
```

@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 2 (Draft). The value is changed to 2 (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 [2.2.1.13](#)) 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 [2.2.1.15](#)) 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.

@Assembly: The fully qualified name of the assembly that implements the web part.

@Class: The name of the .NET class that implements the web part.

@SolutionId: The identifier of the sandboxed solution that implements the web part.

@TheListID: The list identifier of the list associated with the web part.

@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.

@TheFlags: The View Flags, as specified in [MS-WSSFO], section 2.2.2.14, of the Web Part.

@TheType: The web part type identifier of the web part.

@TheBaseViewID: The base view identifier of the web part.

@AllUsersProperties: A binary payload containing 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 binary payload containing 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.

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

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.

Value	Description
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.

3.1.4.5 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            varbinary(max),
    @InstanceDataSize        int,
    @Modifications           nvarchar(max),
    @StatusFieldOrdinal      int,
    @StatusField             nvarchar(64),
    @RequestGuid             uniqueidentifier = NULL OUTPUT
);

```

@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 [2.2.1.11](#), 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 @ListId does not represent a Document Library, this MUST be 1.

@Version: This parameter MUST contain the [list item version](#) of the list Item or the value 0. If the value is 0, the protocol server MUST ignore the list item version. If the value is nonzero and does not match the current list item version value of the list item, the protocol server MUST NOT add the workflow.

@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.

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

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.

Value	Description
1150	The list item version of the list item specified by @ItemId does not match the value of @Version.

Result Sets: MUST NOT return any result sets.

3.1.4.6 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 nvarchar(max),
    @RequestGuid       uniqueidentifier = NULL OUTPUT
);

```

@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 [2.2.1.11](#)) 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 association.

@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, as specified in [\[MS-WSSFO\]](#) section 2.2.2.13, required to manually start any Workflows created from the Workflow association.

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

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

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.

3.1.4.7 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         varbinary(max),  
    @TextPayload           nvarchar(max),  
    @ProcessingId          uniqueidentifier,  
    @AutoDeleteOld         bit = 0,
```

```

        @ExponentialRetryBackOff    bit = 1,
        @RequestGuid                uniqueidentifier = NULL OUTPUT
    );

```

@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. If the parameter is not NULL, then the server MUST schedule the Work Item to run on the given Delivery Date.

@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, when there is no associated Work Item Subtype, the **empty GUID**.

@SiteId: The Site Collection identifier of the Site Collection or, when there is no associated Site Collection, 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 [2.2.1.11](#)) of the Site.

@ItemGuid: The Item GUID of the list Item or, when there is no associated list Item, 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 [2.2.1.13](#)) 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 indicates 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 indicates that the deletion MUST NOT occur. A value of 1 indicates that the deletion MUST occur.

@ExponentialRetryBackOff: This parameter indicates whether or not the server MUST retry execution with exponential backoff from the Work Item Delivery Date when the client calls [proc_RevertInProgressWorkItem](#) or [proc_RevertInProgressWorkItems](#). A value of 0 indicates that exponential backoff MUST NOT occur on retry. A value of 1 indicates that exponential backoff MUST occur on retry.

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

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.

3.1.4.8 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               nvarchar(max),
    @ViewFlags              int OUTPUT,
    @BaseViewId             int OUTPUT,
    @RequestGuid            uniqueidentifier = NULL 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 [2.2.1.15](#)) 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.

@UserId: The User Identifier (as specified in [MS-WSSFO], section [2.2.1.13](#)) 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.

@BaseViewId: This is an output parameter. On output, this is the base view identifier for this Web Part.

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

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.

Value	Description
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.
- 2147467259	An error occurred while the stored procedure was running.

Result Sets: MUST NOT return any result sets.

3.1.4.9 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(
    @RequestGuid          uniqueidentifier = NULL OUTPUT
);
```

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

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

Result Sets: MUST NOT return any result sets.

3.1.4.10 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,
    @RequestGuid     uniqueidentifier = NULL OUTPUT
);
```

@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 ([3.1.1.5](#)) to remove the space used by the deleted Workflows.

@WebId: The Site Identifier (as specified in [\[MS-WSSFO\]](#), section [2.2.1.11](#)) 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.

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

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.

3.1.4.11 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,
    @RequestGuild          uniqueidentifier = NULL OUTPUT
);
```

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

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

Return Code Values: An integer which MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.4.12 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,
    @RequestGuid           uniqueidentifier = NULL OUTPUT
);
```

@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 completed 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.

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

Return Code Values: An integer which MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.4.13 proc_CommitUpdatedZoneIds

The proc_CommitUpdatedZoneIds stored procedure is called to update the webpart zone identifiers of webparts contained in a webpart page. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_CommitUpdatedZoneIds (
    @SiteId            uniqueidentifier,
    @DocId             uniqueidentifier,
    @ZoneIdNew         nvarchar(64),
    @ZoneIdOld         nvarchar(64)
);

```

@SiteId: The Site Collection identifier of the Site Collection which contains the Web Parts to be modified. 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 Parts to be modified. MUST NOT be NULL.

@ZoneIdNew: The webpart zone identifier of the webpart zone containing the web parts to be updated. MUST NOT be NULL.

@ZoneIdOld: The webpart zone identifier to be replaced. MUST NOT be NULL.

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

Result Sets: MUST NOT return any result sets.

3.1.4.14 **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,
    @RequestGuid       uniqueidentifier = NULL OUTPUT
);

```

@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 @SiteId is not NULL, then the server MUST only modify Work Items associated with this Site Collection. If @SiteId 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 @ParentId is not NULL, then the server MUST only modify Work Items which have this Work Item Parent identifier. If @ParentId 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 @BatchId is not NULL, then the server MUST only modify Work Items associated with this Work Item Batch. If @BatchId is NULL, then the server MUST modify Work Items that meet the criteria specified by the other parameters regardless of associated Work Item Batch identifier.

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

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

Result Sets: MUST NOT return any result sets.

3.1.4.15 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,  
    @RequestGuid           uniqueidentifier = NULL OUTPUT  
);
```

@ 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.

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

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.
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.

3.1.4.16 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,  
    @RequestGuid       uniqueidentifier = NULL OUTPUT  
);
```

@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.

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

Return Code Values: An integer which MUST be 0.

Result Sets: MUST return the following result set:

3.1.4.16.1 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.

3.1.4.17 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,  
    @RequestGuid       uniqueidentifier = NULL OUTPUT  
);
```

@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.

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

Return Code Values: An integer which MUST be 0.

Result Sets: MUST return the following result set:

3.1.4.17.1 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.

3.1.4.18 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,  
    @RequestGuid           uniqueidentifier = NULL OUTPUT  
);
```

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

@WebId: The Site Identifier (as specified in [\[MS-WSSFO\]](#), section [2.2.1.11](#)) 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).

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

Return Code Values: An integer which MUST be 0.

Result Sets: MUST return the following result set:

3.1.4.18.1 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.

3.1.4.19 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    varbinary(max),
    @PerUserProperties     varbinary(max),
    @WebPartIdProperty     nvarchar(255),
    @View                  varbinary(max),
    @DisplayName           nvarchar(255),
    @RequestGuid           uniqueidentifier = NULL OUTPUT
);
```

@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. MUST NOT be NULL.

@UserId: The User Identifier (as specified in [\[MS-WSSFO\]](#), section [2.2.1.13](#)) 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 2 (Draft). The value is changed to 2 (Draft) if the Web Part Page is in a Document Library, @Level is 1 (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 binary payload containing zero or more customizable properties on the Web Part.

@PerUserProperties: A binary payload containing zero or more personalizable properties on the Web Part.

@WebPartIdProperty: The optional identifier of the added Web Part. If not NULL, it MUST be unique per page, not begin with a number, and contain only a combination of alphanumeric characters and underscores.

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

@DisplayName: The Display Name for the Web Part.

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

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

Value	Description
0	Successful completion.
1	The List View Web Part was not successfully created.
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 1 (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 1 (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.

3.1.4.20 **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,
    @Synchronization int,
    @Type            int,
    @SequenceNumber  int,
    @Assembly        nvarchar(256),
    @Class           nvarchar(256),
    @Data            nvarchar(256),
    @Filter          nvarchar(256),
    @Credential      int,
    @RequestGuid     uniqueidentifier = NULL OUTPUT
);

```

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

@Id: The event receiver identifier 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 [2.2.1.11](#)) of the site which contains the document.

@ItemId: This parameter MUST be 0.

@Synchronization: The protocol server MUST ignore this parameter.

@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** 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.

@Credential: Reserved. @Credential MUST be 0.

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

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

Value	Description
0	The event receiver identified by @Id was deleted from the site collection identified by @SiteId.
3	The document identified by @DocUrl was 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.

3.1.4.21 `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,
        @RequestGuid         uniqueidentifier = NULL OUTPUT
    );

```

@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 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.

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

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

Value	Description
0	The event receiver whose identifier is @Id and Event Receiver type is not identified by 32767 (as specified in [MS-WSSFO], section 2.2.3.6) was 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 were also deleted.
87	Delete failed because an event receiver was not found whose identifier is @Id or Event Receiver type is not identified by 32767 (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.

3.1.4.22 **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,
    @RequestGuid        uniqueidentifier = NULL OUTPUT
);
```

@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 [2.2.1.11](#)) of the site which contains the event host.

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

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

Return Code Values: An integer which MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.4.23 **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,
    @RequestGuid       uniqueidentifier = NULL OUTPUT
);
```

@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 identifier.

@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.

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

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

Result Sets: MUST NOT return any result sets.

3.1.4.24 **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,  
    @RequestGuid            uniqueidentifier = NULL OUTPUT  
);
```

@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 [2.2.1.13](#)) 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.

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

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.

3.1.4.25 **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,
        @RequestGuid           uniqueidentifier = NULL OUTPUT
    );

```

@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 [2.2.1.13](#)) of the user which is deleting the Web Part specified by @WebPartID.

@WebPartID: The Web Part Identifier (as specified in [\[MS-WSSFO\]](#), section [2.2.1.15](#)) of the Web Part to be deleted. MUST NOT be NULL.

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

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.

Value	Description
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.

3.1.4.26 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,
    @RequestGuid            uniqueidentifier = NULL OUTPUT
);

```

@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 [2.2.1.13](#)) 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 [2.2.1.15](#)) of the Web Part from which to delete personalizations. MUST NOT be NULL.

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

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.

3.1.4.27 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,  
    @RequestGuid            uniqueidentifier = NULL OUTPUT  
) ;
```

@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 [2.2.1.15](#)) of the Web Part to be deleted from the Web Part Page.

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

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

Value	Description
0	Successful completion.
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.

3.1.4.28 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),
```

```

        @RequestGuid                uniqueidentifier = NULL OUTPUT
    );

```

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

@DirName: The Directory Name of the Web Part Page. MUST NOT be NULL.

@LeafName: The Leaf Name of the Web Part Page. MUST NOT be NULL.

@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. MUST NOT be NULL.

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

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.

3.1.4.29 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,
    @RequestGuid            uniqueidentifier = NULL OUTPUT
);

```

@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.

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

Return Code Values: An integer which MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.4.30 **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,  
    @RequestGuid           uniqueidentifier = NULL OUTPUT  
);
```

@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 ([3.1.1.5](#)) 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.

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

Return Code Values: An integer which MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.4.31 **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,  
    @RequestGuid           uniqueidentifier = NULL OUTPUT  
);
```

@SiteId: The Site Collection identifier of the Site Collection which contains the Workflow association. The server MUST update the site collection quota (section [3.1.1.5](#)) 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 an limited, implementation defined 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](#).

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

Return Code Values: An integer which MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.4.32 **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,  
    @RequestGuid     uniqueidentifier = NULL OUTPUT  
);
```

@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.

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

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

Result Sets: MUST NOT return any result sets.

3.1.4.33 **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,  
    @RequestGuid     uniqueidentifier = NULL OUTPUT  
);
```

@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.

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

Return Code Values: An integer which MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.4.34 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,
    @RequestGuid             uniqueidentifier = NULL OUTPUT
);
```

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

@WebId: The Site Identifier (as specified in [\[MS-WSSFO\]](#), section [2.2.1.11](#)) of the Site which contains the list.

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

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

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 [2.2.6.4](#).

3.1.4.35 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,
    @RequestGuid             uniqueidentifier = NULL OUTPUT
);
```

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

@WebId: The Site Identifier, as specified in [\[MS-WSSFO\]](#), section [2.2.1.11](#), of the Site.

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

Return Code Values: An integer which MUST be 0.

Result Sets: MUST return the following result set:

3.1.4.35.1 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_Assembly	nvarchar(255),
tp_Class	nvarchar(255),
tp_SolutionId	uniqueidentifier,
tp_AllUsersProperties	varbinary(max),
tp_PerUserProperties	varbinary(max),
tp_Cache	varbinary(max),
tp_Source	nvarchar(max),
tp_View	nvarchar(max)

tp_ID: The Web Part Identifier, as specified in [\[MS-WSSFO\]](#), section [2.2.1.15](#). 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 [\[MS-WSSFO\]](#), 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: 1 if the Web Part is included the Web Part Page; 0 if the Web Part is not included. 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_Assembly: The fully qualified name of the assembly that implements the web part.

tp_Class: The name of the .NET class that implements the Web Part.

Tp_SolutionId: The identifier of the sandboxed solution or site solution that installed the Web Part.

Tp_AllUsersProperties: A binary payload containing 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 binary payload containing 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 WPV2:WebPart format (as specified in [\[MS-WPPS\]](#), section [2.2.3.2](#)), WPV3:WebPart format (as specified in [\[MS-WPPS\]](#), section [2.2.3.3](#)) or WebParts format (as specified in [\[MS-WPPS\]](#), section [2.2.3.1](#)). The protocol client can determine which format is used by comparing the value against the schemas for the formats. The value will be NULL if the properties are compressed and stored in Tp_AllUserProperties and Tp_PerUserProperties.

Tp_View: The CAML of the Web Part.

3.1.4.36 proc_EnumResourceWarningSites

The proc_EnumResourceWarningSites stored procedure is called to return information about all of the site collections in a content database that have exceeded the warning level for their resource quota. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_EnumResourceWarningSites ();
```

Return Code Values: An integer value which MUST be 0.

Result Sets: This procedure MUST return the Resource Warning Site Collections Result Set.

3.1.4.36.1 Resource Warning Site Collections Result Set

The Resource Warning Site Collections Result Set returns the list of all site collections that have exceeded the warning level for their resource quota. The T-SQL syntax for the result set is as follows:

Id	uniqueidentifier NOT NULL,
BitFlags	int NOT NULL,
CurrentResourceUsage	float NOT NULL,
AverageResourceUsage	float NOT NULL,
ResourceUsageWarning	float NOT NULL,
ResourceUsageMaximum	float NOT NULL;

Id: The site collection identifier of the site collection that has exceeded the warning level for its resource quota.

BitFlags: The Site Collection Flags value, as defined in [\[MS-WSSFO2\]](#), section [2.2.3.9](#), describing the configuration of the site collection.

CurrentResourceUsage: The resource usage value for the site collection during the current monitoring interval for resource usage.

AverageResourceUsage: The mean resource usage value for the site collection over the available number of monitoring intervals for resource usage.

ResourceUsageWarning: The warning level for a resource quota for this site collection.

ResourceUsageMaximum: The maximum level for a resource quota for this site collection.

3.1.4.37 **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,
    @RequestGuid           uniqueidentifier = NULL OUTPUT
);
```

@ProcessingId: The Work Item Processing identifier of the Work Item Process. This parameter MUST NOT be NULL. For each Work Item associated with the given Work Item Processing 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 Processing identifier of the Work Item to NULL.

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

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

Result Sets: MUST NOT return any result sets.

3.1.4.38 **proc_GetAllResourceUsageForSiteToday**

The `proc_GetAllResourceUsageForSiteToday` stored procedure is called to return resource usage values for a given site collection during the current monitoring interval for resource usage. The T-SQL syntax for the stored procedure is as follows:

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

@SiteId: The site collection identifier for the site collection for which resource usage values are requested.

Return Code values: An integer which MUST be 0.

Result Sets: This procedure MUST return the Site Collection Daily Resource Usage Result Set.

3.1.4.38.1 **Site Collection Daily Resource Usage Result Set**

The Site Collection Daily Resource Usage Result Set returns resource usage values for each monitored resource measure reported for the specified site collection over the current monitoring interval for resource usage. The T-SQL syntax for the result set is as follows:

```
ResourceId          uniqueidentifier NOT NULL,
```

```
TotalResourceUsage    float NOT NULL;
```

ResourceId: The identifier of the monitored resource measure for this resource usage value.

TotalResourceUsage: The resource usage value for the monitored resource measure for the specified site collection over the current monitoring interval for resource usage.

3.1.4.39 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               varbinary(512),
    @DirName                nvarchar(256),
    @LeafName               nvarchar(128),
    @Level                  tinyint,
    @PrefetchListScope      bit,
    @ThresholdRowCount      int,
    @GetViewBodies          bit,
    @RequestGuid             uniqueidentifier = NULL OUTPUT
);
```

@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 [\[MS-WSSFO\].proc_FetchDocForHttpGet.Web Parts Metadata, Non-Personalized Result Set](#). If set to 0, Web Parts personalized for the current user are returned in the [\[MS-WSSFO\].proc_FetchDocForHttpGet.WebParts Metadata, Personalized Result Set](#).

@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.

@ThresholdRowCount: The maximum number of rows to return in the List Metadata Result Set.

@GetViewBodies: This value MUST be set to 1.

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

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.

3.1.4.39.1 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.

3.1.4.39.2 Web Parts Metadata, Personalized Result Set

If @AllUsers is 0, [\[MS-WSSFO2\]](#).proc_FetchDocForHttpGet.Web Parts Metadata, Personalized Result Set MUST be returned.

3.1.4.39.3 List Metadata, Result Set

[\[MS-WSSFO\]](#).proc_FetchDocForHttpGet.List Metadata, Result Set MUST be returned.

3.1.4.39.4 List Event Receivers, Result Set

If List Metadata, Result Set is NOT empty then [\[MS-WSSFO\]](#).proc_FetchDocForHttpGet.List Event Receivers, Result Set MUST be returned.

3.1.4.39.5 List Security Information, Result Set

If List Metadata, Result Set is NOT empty then [\[MS-WSSFO\]](#).proc_FetchDocForHttpGet.List Security Information, Result Set MUST be returned.

3.1.4.40 proc_GetAverageDailyResourceUsageForSite

The proc_GetAverageDailyResourceUsageForSite stored procedure is called to return the mean resource usage value of each monitored resource measure for a given site collection. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetAverageDailyResourceUsageForSite (
    @SiteId                uniqueidentifier,
    @RetentionDays         int
);
```

@SiteId: The site collection identifier of the site collection for which to retrieve mean resource usage values.

@RetentionDays: The number of monitoring intervals for which resource usage values are retained.

Return Code values: An integer value which MUST be 0.

Result Sets: This procedure MUST return the Site Collection Average Daily Resource Usage Result Set.

3.1.4.40.1 Site Collection Average Daily Resource Usage Result Set

The Site Collection Average Daily Resource Usage Result Set returns the mean resource usage value for each monitored resource measure for the specified site collection. The T-SQL syntax for the result set is as follows:

```
ResourceId          uniqueidentifier NOT NULL,  
AvgResourceUsage    float NOT NULL;
```

ResourceId: The identifier of a monitored resource measure.

AvgResourceUsage: The mean resource usage value of the monitored resource measure for the specified site collection over the number of monitoring intervals specified by @RetentionDays.

3.1.4.41 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,  
    @RequestGuid            uniqueidentifier = NULL OUTPUT  
);
```

@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.

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

Return Code Values: An integer which MUST be 0.

Result Sets: MUST return the following result set:

3.1.4.41.1 Event Receivers Result Set

The Result Set is defined in the Event Receivers Result Set (as specified in [\[MS-WSSFO2\]](#), section [2.2.6.9](#)). The Result Set MUST also include an additional NULL column at the end of the result set.

3.1.4.42 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,
    @WebId                  uniqueidentifier,
    @HostId                 uniqueidentifier,
    @ContextObjectId        uniqueidentifier,
    @ContextObjectItemId    int,
    @DeleteHostLookupId     uniqueidentifier = NULL,
    @HostType               int = NULL,
    @RequestGuid            uniqueidentifier = NULL OUTPUT
);
```

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

@WebId: The protocol server MUST ignore this value.

@HostId: The event host identifier of the event host for which the event receivers are requested.

@ContextObjectId: The [context object identifier](#) for the Context Object of the Workflow receiver process associated with the Event Receivers to be requested. If this value is not NULL, the protocol server MUST ignore the values of @HostId and @ContextObjectItemId.

@ContextObjectItemId: The context object identifier of the Event Host for which the registered event receivers are requested.

@DeleteHostLookupId: The Event Receiver Identifier (as specified in [\[MS-WSSFO\]](#), section [2.2.1.3](#)) for the Event Receiver to be optionally removed. If the value is NULL, no Event Receivers will 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.

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

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 Event Receiver type is equal to 32767 (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 Event Receiver type is not equal to 32767 (as specified in [\[MS-WSSFO\]](#), section [2.2.3.6](#)), the stored procedure MUST NOT return a result set.

3.1.4.42.1 Event Receivers Result Set

This Result Set will be filtered by the Site through **@SiteId** and the [context object identifier](#) through one of either **@ContextObjectId** or the combination of **@HostId** and **@ContextObjectItemId**. The protocol client MUST specify both the site and the context object. 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 [\[MS-WSSFO2\]](#), section [2.2.6.9](#)), plus one additional NULL column appended to the end of the result set.

3.1.4.43 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),  
    @RequestGuid          uniqueidentifier = NULL OUTPUT  
);
```

@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.

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

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.

3.1.4.43.1 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](#)).

3.1.4.44 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,
@Limit                 int = 0,
@LimitFlags            int = 0,
@RequestGuid           uniqueidentifier = NULL OUTPUT
);

```

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

@WebId: The Site identifier (as specified in [\[MS-WSSFO\]](#), section [2.2.1.11](#)) 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 & @InclusiveFilterState <> 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 & @ExclusiveFilterState <> 0).

@Limit: The optional limit for the number of Workflows returned in the Result Set. This value MUST be a positive integer or 0. If this value is not 0 the server MUST limit the number of returned Workflows in the Result Set to this value. If this value is 0 the server MUST NOT limit the number of returned Workflows in the Result Set.

@LimitFlags: The protocol server MUST ignore this parameter.

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

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.

3.1.4.44.1 List Item Workflows Result Set

The Result Set is defined in section [2.2.6.1](#).

3.1.4.45 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,  
    @RequestGuid            uniqueidentifier = NULL OUTPUT  
);
```

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

@WebId: The Site Identifier (as specified in [\[MS-WSSFO\]](#), section [2.2.1.11](#)) 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.

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

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.

3.1.4.46 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,
    @ViewId                uniqueidentifier,
    @UserID                int,
    @DocVersion            int,
    @bGetAllLevel          bit,
    @bGetDeleted           bit = 0,
    @bGetAllUsers          bit = 0,
    @SiteId                uniqueidentifier,
    @RequestGuid           uniqueidentifier = NULL OUTPUT
);
```

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

@ViewId: The GUID of the list view, or NULL for the default view.

@UserID: The User Identifier (as specified in [\[MS-WSSFO\]](#), section [2.2.1.13](#)) 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.

@DocVersion: The **document version** of the web part page containing the view of the list, or 0 for the current version. MUST NOT be NULL.

@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.

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

Return Code Values: An integer which MUST be 0.

Result Sets: MUST return the following result set:

3.1.4.46.1 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	varbinary(max),
tp_Level	tinyint,
tp_ContentTypeId	varbinary(512),
tp_PageUrlId	uniqueidentifier,
tp_AllUserProperties	varbinary(max),
tp_PerUserProperties	varbinary(max),
tp_WebPartIdProperty	nvarchar(255),
tp_Cache	varbinary(max);

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 in [\[MS-WSSFO\]](#), 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.

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

Tp_AllUsersProperties: A binary payload containing 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 binary payload containing 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_WebPartIdProperty: The optional identifier of the Web Part. If not NULL, it MUST be unique per page, not begin with a number, and contain only a combination of alphanumeric characters and underscores.

Tp_Cache: Private data cache of the Web Part.

3.1.4.47 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,
    @RequestGuid             uniqueidentifier = NULL 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 and @DocID do not reference an existing Web Part Page or @ZoneId does not reference an existing Web Part Zone on the Web Part Page.

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

Return Code Values: An integer which MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.4.48 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),
    @RequestGuid             uniqueidentifier = NULL OUTPUT
);
```

@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 [2.2.1.11](#)) of the site which contains the recycle bin item.

@UserId: The User identifier (as specified in [\[MS-WSSFO\]](#), section [2.2.1.13](#)) of the current user.

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

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

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.

3.1.4.48.1 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.
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.

3.1.4.48.2 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 [\[MS-WSSFO2\]](#), section [2.2.6.9](#)).'

3.1.4.48.3 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 [\[MS-WSSFO2\]](#), section [2.2.6.9](#)).

3.1.4.49 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,
    @RequestGuid           uniqueidentifier = NULL OUTPUT,
);
```

@ProcessingId: The Work Item Processing identifier of the Work Item Process. The server MUST set to this value the Work Item Processing 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

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 both:

- The value of this parameter is not 0, and
- The value of this parameter is less than the number of distinct work item batch identifiers in the set of Work Items matching all of 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.

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

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:

3.1.4.49.1 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 [2.2.6.5](#).

3.1.4.50 proc_GetRunningWorkBatchCount

The proc_GetRunningWorkBatchCount stored procedure is called to retrieve the count of in progress work item batches. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetRunningWorkBatchCount (
    @SiteId                uniqueidentifier,
    @WorkItemType           uniqueidentifier,
    @RequestGuid            uniqueidentifier = NULL OUTPUT
);
```

@SiteId: The Site Collection identifier of the Site Collection. If the parameter is not NULL, then the server MUST only count work item batches associated with this Site Collection. If the parameter is NULL, then the server MUST count work item batches 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. If the parameter is not NULL, then the server MUST only count work item batches associated with this Work Item type. If

the parameter is NULL, then the server MUST count work item batches that meet the criteria specified by the other parameters regardless of Work Item type.

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

Return Code values: An integer value which MUST be the count of the work item batches specified by the @SiteId and @WorkItemType parameters.

Result Sets: MUST NOT return any result sets.

3.1.4.51 **proc_GetSiteResourceUsage**

The proc_GetSiteResourceUsage stored procedure is called to retrieve resource usage values for a site collection. The T-SQL syntax for the stored procedure is as follows:

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

@SiteId: The site collection identifier of the site collection for which the resource usage value is to be retrieved.

Return Code values: An integer value which MUST be 0.

Result Sets: This procedure MUST return the Site Collection Resource Usage Result Set.

3.1.4.51.1 **Site Collection Resource Usage Result Set**

The Site Collection Resource Usage Result Set contains resource usage values for the site collection specified by the @SiteId parameter. The T-SQL syntax for the result set is as follows:

```
CurrentResourceUsage        float NOT NULL,  
AverageResourceUsage        float NOT NULL,  
ResourceUsageMaximum        float NOT NULL;
```

CurrentResourceUsage: The resource usage value for the specified site collection for the current monitoring interval.

AverageResourceUsage: The mean resource usage value for the specified site collection over the available number of monitoring intervals.

ResourceUsageMaximum: The maximum level for resource usage for the specified site collection.

3.1.4.52 **proc_GetSiteSolutionResourceUsage**

The proc_GetSiteSolutionResourceUsage stored procedure is called to retrieve resource usage values for all of the sandboxed solutions in a site collection. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetSiteSolutionResourceUsage (  
    @SiteId                uniqueidentifier,  
    @DaysAgo                int  
) ;
```

@SiteId: The site collection identifier of the site collection for which resource usage values are to be retrieved.

@DaysAgo: The number of monitoring intervals in the past for which resource usage values are to be retrieved. If NULL, the server MUST retrieve resource usage values for all available periods.

Return Code Values: An integer which MUST be 0.

Result Sets: This procedure MUST return the Site Solution Resource Usage Result Set.

3.1.4.52.1 Site Solution Resource Usage Result Set

This result set contains resource usage values for the sandboxed solutions in the specified site collection for the specified monitoring interval. The result set is specified as the Solution Resource Usage Result Set in section [2.2.6.2](#).

3.1.4.53 proc_GetSolutionInfo

The proc_GetSolutionInfo stored procedure is called to get information for the execution of a sandboxed solution. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetSolutionInfo (  
    @SiteId          uniqueidentifier,  
    @SolutionId      uniqueidentifier  
);
```

@SiteId: The site collection identifier of the site collection in which the sandboxed solution resides.

@SolutionId: The identifier of the sandboxed solution.

Return Code Values: An integer which MUST be 0.

Result Sets: This procedure MUST return the Solution Hash Information Result Set.

3.1.4.53.1 Solution Hash Information Result Set

The Solution Hash Information Result Set contains information about the validation state of a sandboxed solution. The T-SQL syntax for the result set is as follows:

```
Hash                nvarchar(50) NOT NULL,  
ValidatorsHash      nvarchar(64) NOT NULL,  
ValidationErrorUrl  nvarchar(4000),  
ValidationErrorMessage nvarchar(4000),  
ResourceQuota       float NOT NULL,  
RecentInvocations   int NOT NULL,  
ResourceQuotaExceeded int NOT NULL;
```

Hash: The implementation-specific hash of the content of the sandboxed solution.

ValidatorsHash: The implementation-specific hash of the validators that validated the sandboxed solution.

ValidationErrorUrl: If the sandboxed solution failed validation, MUST contain the URL with more information about the validation failure.

ValidationErrorMessage: If the sandboxed solution failed validation, MUST contain the specific error message of the validation failure.

ResourceQuota: The resource usage value for the specified sandboxed solution.

RecentInvocations: The number of invocations of code within this sandboxed solution over the current monitoring interval.

ResourceQuotaExceeded: MUST be 1 if the site collection containing the sandboxed solution has exceeded its maximum level for a resource quota, 0 otherwise.

3.1.4.54 **proc_GetSolutionResourceQuota**

The `proc_GetSolutionResourceQuota` stored procedure is called to get the resource usage value for a sandboxed solution. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetSolutionResourceQuota (  
    @SiteId          uniqueidentifier,  
    @SolutionId      uniqueidentifier  
);
```

@SiteId: The site collection identifier of the site collection in which the sandboxed solution resides.

@SolutionId: The identifier of the sandboxed solution.

Return Code Values: An integer which MUST be 0.

Result Sets: This procedure MUST return the Solution Resource Quota Result Set.

3.1.4.54.1 **Solution Resource Quota Result Set**

The Solution Resource Quota Result Set contains the resource usage value for a sandboxed solution. The T-SQL syntax for the result set is as follows:

```
ResourceQuota          float NOT NULL,  
RecentInvocations      int NOT NULL,  
ResourceQuotaExceeded  int NOT NULL;
```

ResourceQuota: The resource usage value for the specified sandboxed solution.

RecentInvocations: The number of invocations of code within this sandboxed solution over the current monitoring interval.

ResourceQuotaExceeded: MUST be 1 if the site collection containing the sandboxed solution has exceeded its maximum level for a resource quota, 0 otherwise.

3.1.4.55 **proc_GetSolutionResourceUsage**

The `proc_GetSolutionResourceUsage` stored procedure is called to get the resource usage values for a sandboxed solution in a specified monitoring interval. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetSolutionResourceUsage (  
    @SiteId          uniqueidentifier,
```

```

        @SolutionId      uniqueidentifier,
        @DaysAgo          int
    );

```

@SiteId: The site collection identifier of the site collection in which the sandboxed solution resides.

@SolutionId: The identifier of the sandboxed solution.

@DaysAgo: The number of monitoring intervals in the past for which resource usage values are to be retrieved. If NULL, the server MUST retrieve resource usage values for all available monitoring intervals.

Return Code Values: An integer which MUST be 0.

Result Sets: This procedure MUST return the Solution Resource Usage Result Set.

3.1.4.55.1 Solution Resource Usage Result Set

This result set contains resource usage values for the specified sandboxed solution in the specified site collection for the specified monitoring interval. The result set is specified in section [2.2.6.2](#).

3.1.4.56 proc_GetSolutionResourceUsageDailyOrdinal

The `proc_GetSolutionResourceUsageDailyOrdinal` stored procedure is called to get the ordinal for the current monitoring interval for resource usage. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_GetSolutionResourceUsageDailyOrdinal ();

```

Return Code Values: An integer which MUST be 0.

Result Sets: This procedure MUST return the Solution Resource Usage Daily Ordinal Result Set.

3.1.4.56.1 Solution Resource Usage Daily Ordinal Result Set

The Solution Resource Usage Daily Ordinal Result set MUST contain one row containing the ordinal for the current monitoring interval for resource usage. The T-SQL syntax for the result set is as follows:

```

DaysAgo          int NOT NULL;

```

DaysAgo: The ordinal for the current monitoring interval for resource usage..

3.1.4.57 proc_GetSolutionsData

The `proc_GetSolutionsData` stored procedure is called to get information for all sandboxed solutions in a site collection. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_GetSolutionsData (
    @SiteId          uniqueidentifier
);

```

@SiteId: The site collection identifier of the site collection in which the sandboxed solutions reside.

Return Code Values: An integer value which MUST be 0.

Result Sets: This procedure MUST return the Solution Data Result Set.

3.1.4.57.1 Solution Data Result Set

The Solution Data Result Set contains information about the sandboxed solutions in a site collection. The T-SQL syntax for the result set is as follows:

Name	nvarchar(128) NOT NULL,
SolutionId	uniqueidentifier NOT NULL,
Hash	nvarchar(50) NOT NULL,
Status	smallint NOT NULL,
HasAssemblies	tinyint NOT NULL,
Definitions	varbinary(max) NOT NULL;

Name: The name of the sandboxed solution.

SolutionId: The identifier of the sandboxed solution.

Hash: The implementation-specific hash of the content of the sandboxed solution.

Status: A [Sandboxed Solution Status value](#) corresponding to the status of the sandboxed solution.

HasAssemblies: MUST be 1 if the sandboxed solution contains assemblies. 0 otherwise.

Definitions: The implementation-specific serialization of the feature definitions for the sandboxed solution.

3.1.4.58 proc_GetWFTemplatesLastModifiedForWeb

The proc_GetWFTemplatesLastModifiedForWeb stored procedure is called to retrieve modification information about document libraries contained in a site collection and site. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetWFTemplatesLastModifiedForWeb (  
    @SiteId                uniqueidentifier,  
    @WebId                 uniqueidentifier,  
    @SiteRootLastModified  datetime OUTPUT,  
    @CurWebLastModified    datetime OUTPUT,  
    @SiteRootItemCount     int OUTPUT,  
    @CurWebItemCount       int OUTPUT,  
    @RequestGuid            uniqueidentifier = NULL OUTPUT  
);
```

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

@WebId: The Site identifier (as specified in [\[MS-WSSFO\]](#), section [2.2.1.11](#)) of the Site which contains the document libraries. This value MUST NOT be NULL.

@SiteRootLastModified: The protocol server MUST ignore the input value of this parameter. On output, the protocol server MUST set this value to the latest modification time of any document

library based on template 122 (No Code Public) contained in the Site Collection specified by @SiteId, or NULL if there are no document libraries matching the criteria.

@CurWebLastModified: The protocol server MUST ignore the input value of this parameter. On output, the protocol server MUST set this value to the latest modification time of any document library based on template 117 (No Code Workflows) contained in the Site specified by @WebId, or NULL if there are no document libraries matching the criteria.

@SiteRootItemCount: The protocol server MUST ignore the input value of this parameter. On output, the protocol server MUST set this value to the total count of items contained in all document libraries based on template 122 (No Code Public) contained in the Site Collection specified by @SiteId, or NULL if there are no document libraries matching the criteria.

@CurWebItemCount: The protocol server MUST ignore the input value of this parameter. On output, the protocol server MUST set this value to the total count of items contained in all document libraries based on template 117 (No Code Workflows) contained in the Site specified by @WebId, or NULL if there are no document libraries matching the criteria.

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

3.1.4.59 **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),
    @RequestGuid           uniqueidentifier = NULL OUTPUT
);
```

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

@WebId: The Site identifier (as specified in [\[MS-WSSFO\]](#), section [2.2.1.11](#)) 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.

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

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

Result Sets: MUST return the [Workflow Associations Result Set](#).

3.1.4.59.1 Workflow Associations Result Set

The Result Set is defined in section [2.2.6.5](#).

3.1.4.60 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,  
    @RequestGuid             uniqueidentifier = NULL OUTPUT  
);
```

@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 [2.2.1.11](#)) 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 list items.

@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.

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

Return Code Values: An integer which MUST be 0.

Result Set: MUST return 0, 1, or 2 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).

3.1.4.60.1 Workflow Associations Result Set

The Result Set is defined in [2.2.6.5](#).

3.1.4.60.2 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](#).

3.1.4.61 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,
    @RequestGuid           uniqueidentifier = NULL OUTPUT
);
```

@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.

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

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

Result Sets: MUST return the following result set:

3.1.4.61.1 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 [2.2.6.5](#).

3.1.4.61.2 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 [2.2.6.5](#).

3.1.4.62 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,
    @Synchronization   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,
    @RequestGuid       uniqueidentifier = NULL OUTPUT
);
```

@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 [2.2.1.11](#)) 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](#)).

@Synchronization: Specifies the synchronicity of the event receiver and the action triggering the event. The value MUST be an integer which is listed in the following table:

Value	Name	Description
0	Default	For before event receivers , the server MUST run the event receiver synchronously. For after event receivers , the server is not required to run the event synchronously.
1	Synchronous	The server MUST run the event receiver using the same thread that is processing the request whose action triggered the event.
2	Asynchronous	The server MUST queue the task of running the event receiver. The server is not required to run the task using the same thread that is processing the request whose action triggered the event.

@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 of the Event Receiver. @SequenceNumber MUST be greater than or equal to 0 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 0.

@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.

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

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

Value	Description
0	Successful completion.
30	An error occurred.
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.

3.1.4.63 **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,
    @Synchronization int,
    @Type            int,
    @SequenceNumber  int,
    @Assembly        nvarchar(256),
    @Class           nvarchar(256),
    @SolutionId      uniqueidentifier,
    @Data            nvarchar(256),
    @Filter          nvarchar(256),
    @Credential      int,
    @RequestGuid     uniqueidentifier = NULL OUTPUT
);
```

@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 [2.2.1.11](#)) of the site which contains the document.

@ItemId: Reserved. @ItemId MUST be 0.

@Synchronization: Specifies the synchronicity of the event receiver and the action triggering the event. The value MUST be an integer which is listed in the following table:

Value	Name	Description
0	Default	For before event receivers, the server MUST run the event receiver synchronously. For after event receivers, the server is not required to run the event synchronously.
1	Synchronous	The server MUST run the event receiver using the same thread that is processing the request whose action triggered the event.
2	Asynchronous	The server MUST queue the task of running the event receiver. The server is not required to run the task using the same thread that is processing the request whose action triggered the event.

@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 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. 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.

@SolutionId: The identifier of the sandboxed solution that is the source 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.

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

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.

3.1.4.64 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),
    @Synchronization   int,
    @Type              int,
    @SequenceNumber    int,
    @Assembly          nvarchar(256),
    @Class             nvarchar(256),
    @SolutionId        uniqueidentifier,
    @Data              nvarchar(256),
    @Filter            nvarchar(256),
    @SourceId          varbinary(512),
    @SourceType        int,
    @Credential        int,
    @ContextType       uniqueidentifier,
    @ContextEventType  uniqueidentifier,
    @ContextId         uniqueidentifier,
    @ContextObjectId   uniqueidentifier,
    @ContextCollectionId uniqueidentifier,
    @RequestGuid       uniqueidentifier = NULL OUTPUT
);
```

@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 [2.2.1.11](#)) of the site which contains the event host. If there is no associated Site, then this value MUST be the empty GUID. 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.

@Synchronization: Specifies the synchronicity of the event receiver and the action triggering the event. The value MUST be an integer which is listed in the following table:

Value	Name	Description
0	Default	For before event receivers, the server MUST run the event receiver synchronously. For after event receivers, the server is not required to run the event synchronously.
1	Synchronous	The server MUST run the event receiver synchronously.
2	Asynchronous	The server MUST queue the task of running the event receiver. The server is not required to run the task synchronously.

@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 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.

@SolutionId: The identifier of the sandboxed solution.

@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.

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

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

Value	Description
0	Insertion succeeded.
30	An error occurred.
87	Insertion failed.

Result Sets: MUST NOT return any result sets.

3.1.4.65 **proc_LogSolutionResourceUsage20**

The `proc_LogSolutionResourceUsage20` stored procedure is called to log to the immediate solution resource usage log up to 20 resource usage measurements for sandboxed solutions within a given site collection. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_LogSolutionResourceUsage20 (
    @SiteId                                uniqueidentifier,
    @SolutionId01                          uniqueidentifier = null,
    @ResourceId01                          uniqueidentifier = null,
    @StartTime01                           datetime = null,
    @EndTime01                             datetime = null,
    @SampleCount01                         int = null,
    @ResourceUsage01                       float = null,
    @SolutionId02                          uniqueidentifier = null,
    @ResourceId02                          uniqueidentifier = null,
    @StartTime02                           datetime = null,
    @EndTime02                             datetime = null,
    @SampleCount02                         int = null,
    @ResourceUsage02                       float = null,
    @SolutionId03                          uniqueidentifier = null,
    @ResourceId03                          uniqueidentifier = null,
    @StartTime03                           datetime = null,
    @EndTime03                             datetime = null,
    @SampleCount03                         int = null,
    @ResourceUsage03                       float = null,
    @SolutionId04                          uniqueidentifier = null,
    @ResourceId04                          uniqueidentifier = null,
    @StartTime04                           datetime = null,
    @EndTime04                             datetime = null,
    @SampleCount04                         int = null,
    @ResourceUsage04                       float = null,
    @SolutionId05                          uniqueidentifier = null,
    @ResourceId05                          uniqueidentifier = null,
    @StartTime05                           datetime = null,
    @EndTime05                             datetime = null,
    @SampleCount05                         int = null,
    @ResourceUsage05                       float = null,
    @SolutionId06                          uniqueidentifier = null,
    @ResourceId06                          uniqueidentifier = null,
    @StartTime06                           datetime = null,
    @EndTime06                             datetime = null,
    @SampleCount06                         int = null,
    @ResourceUsage06                       float = null,
    @SolutionId07                          uniqueidentifier = null,
    @ResourceId07                          uniqueidentifier = null,
    @StartTime07                           datetime = null,
    @EndTime07                             datetime = null,

```

@SampleCount07	int = null,
@ResourceUsage07	float = null,
@SolutionId08	uniqueidentifier = null,
@ResourceId08	uniqueidentifier = null,
@StartTime08	datetime = null,
@EndTime08	datetime = null,
@SampleCount08	int = null,
@ResourceUsage08	float = null,
@SolutionId09	uniqueidentifier = null,
@ResourceId09	uniqueidentifier = null,
@StartTime09	datetime = null,
@EndTime09	datetime = null,
@SampleCount09	int = null,
@ResourceUsage09	float = null,
@SolutionId10	uniqueidentifier = null,
@ResourceId10	uniqueidentifier = null,
@StartTime10	datetime = null,
@EndTime10	datetime = null,
@SampleCount10	int = null,
@ResourceUsage10	float = null,
@SolutionId11	uniqueidentifier = null,
@ResourceId11	uniqueidentifier = null,
@StartTime11	datetime = null,
@EndTime11	datetime = null,
@SampleCount11	int = null,
@ResourceUsage11	float = null,
@SolutionId12	uniqueidentifier = null,
@ResourceId12	uniqueidentifier = null,
@StartTime12	datetime = null,
@EndTime12	datetime = null,
@SampleCount12	int = null,
@ResourceUsage12	float = null,
@SolutionId13	uniqueidentifier = null,
@ResourceId13	uniqueidentifier = null,
@StartTime13	datetime = null,
@EndTime13	datetime = null,
@SampleCount13	int = null,
@ResourceUsage13	float = null,
@SolutionId14	uniqueidentifier = null,
@ResourceId14	uniqueidentifier = null,
@StartTime14	datetime = null,
@EndTime14	datetime = null,
@SampleCount14	int = null,
@ResourceUsage14	float = null,
@SolutionId15	uniqueidentifier = null,
@ResourceId15	uniqueidentifier = null,
@StartTime15	datetime = null,
@EndTime15	datetime = null,
@SampleCount15	int = null,
@ResourceUsage15	float = null,
@SolutionId16	uniqueidentifier = null,
@ResourceId16	uniqueidentifier = null,
@StartTime16	datetime = null,
@EndTime16	datetime = null,
@SampleCount16	int = null,
@ResourceUsage16	float = null,
@SolutionId17	uniqueidentifier = null,
@ResourceId17	uniqueidentifier = null,
@StartTime17	datetime = null,

```

@EndTime17          datetime = null,
@SampleCount17      int = null,
@ResourceUsage17    float = null,
@SolutionId18       uniqueidentifier = null,
@ResourceId18        uniqueidentifier = null,
@StartTime18        datetime = null,
@EndTime18          datetime = null,
@SampleCount18      int = null,
@ResourceUsage18    float = null,
@SolutionId19       uniqueidentifier = null,
@ResourceId19        uniqueidentifier = null,
@StartTime19        datetime = null,
@EndTime19          datetime = null,
@SampleCount19      int = null,
@ResourceUsage19    float = null,
@SolutionId20       uniqueidentifier = null,
@ResourceId20        uniqueidentifier = null,
@StartTime20        datetime = null,
@EndTime20          datetime = null,
@SampleCount20      int = null,
@ResourceUsage20    float = null
);

```

@SiteId: The site collection identifier of the site collection for this resource usage measurement.

The next six parameters are duplicated 20 times, with each set of parameters referring to a resource usage measurement to be logged. Each instance of these individual parameter names is differentiated by a suffix with a value of 01 through 20 inclusive, which replaces the placeholder "#" symbol shown following. Each group is optional. To signify that a group is to be ignored, the client MUST specify a **@ResourceId#** value of NULL, in which case the server MUST ignore the other parameters with that suffix value.

@SolutionId#: The identifier of the sandboxed solution for this resource usage measurement.

@ResourceId#: The identifier of the monitored resource measure for this resource usage measurement.

@StartTime#: The beginning of the time interval, in the local time zone of the front-end Web server, corresponding to this resource usage measurement.

@EndTime#: The end of the time interval, in the local time zone of the front-end Web server, corresponding to this resource usage measurement.

@SampleCount#: The number of sample points taken for this resource usage measurement.

@ResourceUsage#: The resource usage measurement for the given sandboxed solution, monitored resource measure, and time interval.

Return Code Values: An integer value which MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.4.66 **proc_LogSolutionResourceUsageDaily20**

The `proc_LogSolutionResourceUsageDaily20` stored procedure is called to log to the daily solution resource usage log up to 20 aggregated daily resource usage measurements for sandboxed solutions within a given site collection. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_LogSolutionResourceUsageDaily20 (
    @SiteId01            uniqueidentifier = null,
    @SolutionId01        uniqueidentifier = null,
    @ResourceId01        uniqueidentifier = null,
    @SampleCount01       int = null,
    @ResourceUsage01     float = null,
    @SiteId02            uniqueidentifier = null,
    @SolutionId02        uniqueidentifier = null,
    @ResourceId02        uniqueidentifier = null,
    @SampleCount02       int = null,
    @ResourceUsage02     float = null,
    @SiteId03            uniqueidentifier = null,
    @SolutionId03        uniqueidentifier = null,
    @ResourceId03        uniqueidentifier = null,
    @SampleCount03       int = null,
    @ResourceUsage03     float = null,
    @SiteId04            uniqueidentifier = null,
    @SolutionId04        uniqueidentifier = null,
    @ResourceId04        uniqueidentifier = null,
    @SampleCount04       int = null,
    @ResourceUsage04     float = null,
    @SiteId05            uniqueidentifier = null,
    @SolutionId05        uniqueidentifier = null,
    @ResourceId05        uniqueidentifier = null,
    @SampleCount05       int = null,
    @ResourceUsage05     float = null,
    @SiteId06            uniqueidentifier = null,
    @SolutionId06        uniqueidentifier = null,
    @ResourceId06        uniqueidentifier = null,
    @SampleCount06       int = null,
    @ResourceUsage06     float = null,
    @SiteId07            uniqueidentifier = null,
    @SolutionId07        uniqueidentifier = null,
    @ResourceId07        uniqueidentifier = null,
    @SampleCount07       int = null,
    @ResourceUsage07     float = null,
    @SiteId08            uniqueidentifier = null,
    @SolutionId08        uniqueidentifier = null,
    @ResourceId08        uniqueidentifier = null,
    @SampleCount08       int = null,
    @ResourceUsage08     float = null,
    @SiteId09            uniqueidentifier = null,
    @SolutionId09        uniqueidentifier = null,
    @ResourceId09        uniqueidentifier = null,
    @SampleCount09       int = null,
    @ResourceUsage09     float = null,
    @SiteId10            uniqueidentifier = null,
    @SolutionId10        uniqueidentifier = null,
    @ResourceId10        uniqueidentifier = null,
    @SampleCount10       int = null,
    @ResourceUsage10     float = null,
    @SiteId11            uniqueidentifier = null,
    @SolutionId11        uniqueidentifier = null,
    @ResourceId11        uniqueidentifier = null,
    @SampleCount11       int = null,
    @ResourceUsage11     float = null,
    @SiteId12            uniqueidentifier = null,
    @SolutionId12        uniqueidentifier = null,
    @ResourceId12        uniqueidentifier = null,

```

```

@SampleCount12      int = null,
@ResourceUsage12     float = null,
@SiteId13            uniqueidentifier = null,
@SolutionId13        uniqueidentifier = null,
@ResourceId13         uniqueidentifier = null,
@SampleCount13       int = null,
@ResourceUsage13      float = null,
@SiteId14            uniqueidentifier = null,
@SolutionId14         uniqueidentifier = null,
@ResourceId14         uniqueidentifier = null,
@SampleCount14       int = null,
@ResourceUsage14      float = null,
@SiteId15            uniqueidentifier = null,
@SolutionId15         uniqueidentifier = null,
@ResourceId15         uniqueidentifier = null,
@SampleCount15       int = null,
@ResourceUsage15      float = null,
@SiteId16            uniqueidentifier = null,
@SolutionId16         uniqueidentifier = null,
@ResourceId16         uniqueidentifier = null,
@SampleCount16       int = null,
@ResourceUsage16      float = null,
@SiteId17            uniqueidentifier = null,
@SolutionId17         uniqueidentifier = null,
@ResourceId17         uniqueidentifier = null,
@SampleCount17       int = null,
@ResourceUsage17      float = null,
@SiteId18            uniqueidentifier = null,
@SolutionId18         uniqueidentifier = null,
@ResourceId18         uniqueidentifier = null,
@SampleCount18       int = null,
@ResourceUsage18      float = null,
@SiteId19            uniqueidentifier = null,
@SolutionId19         uniqueidentifier = null,
@ResourceId19         uniqueidentifier = null,
@SampleCount19       int = null,
@ResourceUsage19      float = null,
@SiteId20            uniqueidentifier = null,
@SolutionId20         uniqueidentifier = null,
@ResourceId20         uniqueidentifier = null,
@SampleCount20       int = null,
@ResourceUsage20      float = null
);

```

The next five parameters are duplicated 20 times, with each set of parameters referring to a aggregated daily resource usage measurement to be logged. Each instance of these individual parameter names is differentiated by a suffix with a value of 01 through 20 inclusive, which replaces the placeholder "#" symbol shown following. Each group is optional. To signify that a group is to be ignored, the client MUST specify a **@ResourceId#** value of NULL, in which case the server MUST ignore the other parameters with that suffix value.

@SiteId#: The site collection identifier of the site collection for this resource usage measurement.

@SolutionId#: The identifier of the sandboxed solution for this resource usage measurement.

@ResourceId#: The identifier of the monitored resource measure for this resource usage measurement.

@SampleCount#: The number of sample points taken for this resource usage measurement.

@ResourceUsage#: The resource usage measurement for the given site collection, sandboxed solution and monitored resource measure.

Return Code Values: An integer value which MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.4.67 **proc_LogSolutionResourceUsageWindowed20**

The `proc_LogSolutionResourceUsageWindowed20` stored procedure is called to log to the windowed solution resource usage log up to 20 resource usage measurements for sandboxed solution. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_LogSolutionResourceUsageWindowed20 (
    @DaysAgo                int,
    @SiteId01                uniqueidentifier = null,
    @SolutionId01            uniqueidentifier = null,
    @ResourceId01            uniqueidentifier = null,
    @StartTime01             datetime = null,
    @EndTime01              datetime = null,
    @SampleCount01           int = null,
    @ResourceUsage01         float = null,
    @SiteId02                uniqueidentifier = null,
    @SolutionId02            uniqueidentifier = null,
    @ResourceId02            uniqueidentifier = null,
    @StartTime02             datetime = null,
    @EndTime02              datetime = null,
    @SampleCount02           int = null,
    @ResourceUsage02         float = null,
    @SiteId03                uniqueidentifier = null,
    @SolutionId03            uniqueidentifier = null,
    @ResourceId03            uniqueidentifier = null,
    @StartTime03             datetime = null,
    @EndTime03              datetime = null,
    @SampleCount03           int = null,
    @ResourceUsage03         float = null,
    @SiteId04                uniqueidentifier = null,
    @SolutionId04            uniqueidentifier = null,
    @ResourceId04            uniqueidentifier = null,
    @StartTime04             datetime = null,
    @EndTime04              datetime = null,
    @SampleCount04           int = null,
    @ResourceUsage04         float = null,
    @SiteId05                uniqueidentifier = null,
    @SolutionId05            uniqueidentifier = null,
    @ResourceId05            uniqueidentifier = null,
    @StartTime05             datetime = null,
    @EndTime05              datetime = null,
    @SampleCount05           int = null,
    @ResourceUsage05         float = null,
    @SiteId06                uniqueidentifier = null,
    @SolutionId06            uniqueidentifier = null,
    @ResourceId06            uniqueidentifier = null,
    @StartTime06             datetime = null,
    @EndTime06              datetime = null,
    @SampleCount06           int = null,
```

@ResourceUsage06	float = null,
@SiteId07	uniqueidentifier = null,
@SolutionId07	uniqueidentifier = null,
@ResourceId07	uniqueidentifier = null,
@StartTime07	datetime = null,
@EndTime07	datetime = null,
@SampleCount07	int = null,
@ResourceUsage07	float = null,
@SiteId08	uniqueidentifier = null,
@SolutionId08	uniqueidentifier = null,
@ResourceId08	uniqueidentifier = null,
@StartTime08	datetime = null,
@EndTime08	datetime = null,
@SampleCount08	int = null,
@ResourceUsage08	float = null,
@SiteId09	uniqueidentifier = null,
@SolutionId09	uniqueidentifier = null,
@ResourceId09	uniqueidentifier = null,
@StartTime09	datetime = null,
@EndTime09	datetime = null,
@SampleCount09	int = null,
@ResourceUsage09	float = null,
@SiteId10	uniqueidentifier = null,
@SolutionId10	uniqueidentifier = null,
@ResourceId10	uniqueidentifier = null,
@StartTime10	datetime = null,
@EndTime10	datetime = null,
@SampleCount10	int = null,
@ResourceUsage10	float = null,
@SiteId11	uniqueidentifier = null,
@SolutionId11	uniqueidentifier = null,
@ResourceId11	uniqueidentifier = null,
@StartTime11	datetime = null,
@EndTime11	datetime = null,
@SampleCount11	int = null,
@ResourceUsage11	float = null,
@SiteId12	uniqueidentifier = null,
@SolutionId12	uniqueidentifier = null,
@ResourceId12	uniqueidentifier = null,
@StartTime12	datetime = null,
@EndTime12	datetime = null,
@SampleCount12	int = null,
@ResourceUsage12	float = null,
@SiteId13	uniqueidentifier = null,
@SolutionId13	uniqueidentifier = null,
@ResourceId13	uniqueidentifier = null,
@StartTime13	datetime = null,
@EndTime13	datetime = null,
@SampleCount13	int = null,
@ResourceUsage13	float = null,
@SiteId14	uniqueidentifier = null,
@SolutionId14	uniqueidentifier = null,
@ResourceId14	uniqueidentifier = null,
@StartTime14	datetime = null,
@EndTime14	datetime = null,
@SampleCount14	int = null,
@ResourceUsage14	float = null,
@SiteId15	uniqueidentifier = null,
@SolutionId15	uniqueidentifier = null,

```

@ResourceId15      uniqueidentifier = null,
@StartTime15       datetime = null,
@EndTime15         datetime = null,
@SampleCount15     int = null,
@ResourceUsage15    float = null,
@SiteId16           uniqueidentifier = null,
@SolutionId16       uniqueidentifier = null,
@ResourceId16       uniqueidentifier = null,
@StartTime16        datetime = null,
@EndTime16          datetime = null,
@SampleCount16      int = null,
@ResourceUsage16     float = null,
@SiteId17           uniqueidentifier = null,
@SolutionId17        uniqueidentifier = null,
@ResourceId17        uniqueidentifier = null,
@StartTime17         datetime = null,
@EndTime17           datetime = null,
@SampleCount17       int = null,
@ResourceUsage17      float = null,
@SiteId18           uniqueidentifier = null,
@SolutionId18        uniqueidentifier = null,
@ResourceId18        uniqueidentifier = null,
@StartTime18         datetime = null,
@EndTime18           datetime = null,
@SampleCount18       int = null,
@ResourceUsage18      float = null,
@SiteId19           uniqueidentifier = null,
@SolutionId19        uniqueidentifier = null,
@ResourceId19        uniqueidentifier = null,
@StartTime19         datetime = null,
@EndTime19           datetime = null,
@SampleCount19       int = null,
@ResourceUsage19      float = null,
@SiteId20           uniqueidentifier = null,
@SolutionId20        uniqueidentifier = null,
@ResourceId20        uniqueidentifier = null,
@StartTime20         datetime = null,
@EndTime20           datetime = null,
@SampleCount20       int = null,
@ResourceUsage20      float = null
);

```

@DaysAgo: The ordinal for the monitoring interval associated with these resource usage measurements. If NULL, the server MUST use the current ordinal for the monitoring interval for resource usage.

The next seven parameters are duplicated 20 times, with each set of parameters referring to a resource usage measurement to be logged. Each instance of these individual parameter names is differentiated by a suffix with a value of 01 through 20 inclusive, which replaces the placeholder "#" symbol shown following. Each group is optional. To signify that a group is to be ignored, the client MUST specify a **@ResourceId#** value of NULL, in which case the server MUST ignore the other parameters with that suffix value.

@SiteId#: The site collection identifier of the site collection for this resource usage measurement.

@SolutionId#: The identifier of the sandboxed solution for this resource usage measurement.

@ResourceId#: The identifier of the monitored resource measure for this resource usage measurement.

@StartTime#: The beginning of the time interval, in the local time zone of the front-end Web server, corresponding to this resource usage measurement.

@EndTime#: The end of the time interval, in the local time zone of the front-end Web server, corresponding to this resource usage measurement.

@SampleCount#: The number of sample points taken for this resource usage measurement.

@ResourceUsage#: The resource usage measurement for the given sandboxed solution, monitored resource measure, and time interval.

Return Code Values: An integer value which MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.4.68 **proc_ProcessSolutionResourceUsageLogData**

The **proc_ProcessSolutionResourceUsageLogData** stored procedure is called to read resource usage measurements from the immediate solution resource usage log in batches for aggregation. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ProcessSolutionResourceUsageLogData (  
    @IdStart          bigint  
) ;
```

@IdStart: The identifier of the resource usage measurement for the record with which this batch is to start. If NULL, the server MUST send the resource usage measurements with the largest identifiers, up to 1001 resource usage measurements. Otherwise, the server MUST send the resource usage measurements with the largest identifiers strictly less than @IdStart, up to 1001 resource usage measurements.

Return Code Values: An integer value which MUST be 0.

Result Sets: This procedure MUST return the Solution Resource Usage Log Processing Result Set.

3.1.4.68.1 **Solution Resource Usage Log Processing Result Set**

This result set MUST contain from 0 up to 1001 rows containing resource usage measurements from the immediate solution resource usage log. The **result set** is defined in the Solution Resource Usage Processing Result Set (as specified in section [2.2.6.3](#)).

3.1.4.69 **proc_ProcessSolutionResourceUsageWindowedData**

The **proc_ProcessSolutionResourceUsageWindowedData** stored procedure is called to read resource usage measurements from the windowed solution resource usage log in batches for aggregation. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ProcessSolutionResourceUsageWindowedData (  
    @IdStart          bigint,  
    @DaysAgo          int  
) ;
```

@IdStart: The identifier for the resource usage measurement record with which this batch is to start. If NULL, the server MUST send the resource usage measurements with the largest identifiers, up to 1001 resource usage measurements. Otherwise, the server MUST send the resource usage measurements with the largest identifiers strictly less than @IdStart, up to 1001 resource usage measurements.

@DaysAgo: The ordinal for the monitoring interval for resource usage for which records are being retrieved.

Return Code Values: An integer value which MUST be 0.

Result Sets: This procedure MUST return the Windowed Solution Resource Usage Processing Result Set.

3.1.4.69.1 Windowed Solution Resource Usage Processing Result Set

This result set MUST contain 0 to 1001 rows containing resource usage measurements from the windowed solution resource usage log. The **result set** is defined in the Solution Resource Usage Processing Result Set (as specified in section [2.2.6.3](#)).

3.1.4.70 proc_ProvisionWebPart

The proc_ProvisionWebPart stored procedure is called to add a new Web Part to a Web Part Page. 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                nvarchar(max),
    @RequestGuid           uniqueidentifier = NULL OUTPUT
);
```

@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. 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. 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 [2.2.3.2](#)) or HTML format.

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

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.

3.1.4.71 **proc_RemoveSolution**

The `proc_RemoveSolution` stored procedure is called to remove a sandboxed solution from the specified site collection. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_RemoveSolution (  
    @SiteId          uniqueidentifier,  
    @SolutionId      uniqueidentifier,  
    @Hash            nvarchar(50)  
) ;
```

@SiteId: The site collection identifier of the site collection from which the sandboxed solution is to be removed. This value MUST NOT be NULL.

@SolutionId: The identifier of the sandboxed solution. This value MUST NOT be NULL.

@Hash: The implementation-specific hash of the content of the sandboxed solution. This value MUST NOT be NULL.

Return Code Values: An integer value which MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.4.72 **proc_ResetSiteResourceUsageWarnings**

The `proc_ResetSiteResourceUsageWarnings` stored procedure is called to do a bulk reset of the resource quota flags for this Site Collection. See [3.1.1.5 Quota Management](#) for more information. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ResetSiteResourceUsageWarnings () ;
```

Return Code Values: An integer value which MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.4.73 **proc_RestoreWebPartForDoc**

The **proc_RestoreWebPartForDoc** stored procedure is called to restore Web Parts to a Web Part Page while restoring the Web Part Page to a previous version.

```
PROCEDURE proc_RestoreWebPartForDoc (
    @SiteId            uniqueidentifier,
    @DirName           nvarchar(256),
    @LeafName          nvarchar(128),
    @Level             tinyint,
    @OldVersion        int,
    @RequestGuid       uniqueidentifier = null OUTPUT
);
```

@SiteId: The Site Collection identifier of the Site Collection containing the Web Part Page. MUST NOT be NULL.

@DirName: The Directory Name of the Web Part Page. MUST NOT be NULL.

@LeafName: The Leaf Name of the Web Part Page. MUST NOT be NULL.

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

@OldVersion: The version of the Web Part Page being restored. MUST NOT be NULL.

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

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

Value	Description
0	Successful completion.
2	The Web Part Page was not found.
-2147467259	An error occurred while the stored procedure was running.

Result Sets: MUST NOT return any result sets.

3.1.4.74 **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 backoff on the Work Item Delivery Date; exponential backoff only occurs if the Work Item in question is marked for exponential backoff. 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,
    @RequestGuid       uniqueidentifier = NULL OUTPUT
);
```

);

@ProcessingId: The Work Item Processing 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.

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

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

Result Sets: MUST NOT return any result sets.

3.1.4.75 **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 backoff on the Work Item Delivery Date; exponential backoff only occurs if the Work Item in question is marked for exponential backoff. 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,
    @RequestGuid           uniqueidentifier = NULL OUTPUT
);
```

@ProcessingId: The Work Item Processing 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.

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

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

Result Sets: MUST NOT return any result sets.

3.1.4.76 **proc_SetEventReceiverToSynchronous**

The **proc_SetEventReceiverToSynchronous** stored procedure is called to set all registrations of the given event receiver to synchronous. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_SetEventReceiverToSynchronous (
    @SiteId          uniqueidentifier,
    @Assembly         nvarchar(256),
    @Class            nvarchar(256),
    @Type             int
);
```

@SiteId: The site collection identifier of the site collection. This value MUST NOT be NULL.

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

@Class: The fully qualified class name of the implementation of the event receiver.

@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](#)).

Return Code Values: An integer value which MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.4.77 **proc_TruncateResourceUsageDaily**

The **proc_TruncateResourceUsageDaily** stored procedure is called to remove resource usage measurements from the daily solution resource usage log. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_TruncateResourceUsageDaily (
    @IdStart bigint
);
```

@IdStart: The identifier of the resource usage measurements at which removal of records will begin. The server MUST remove all resource usage measurements from the daily solution resource usage log whose identifier is less than or equal to the specified value.

Return Code Values: An integer value which MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.4.78 **proc_TruncateResourceUsageLog**

The **proc_TruncateResourceUsageLog** stored procedure is called to remove resource usage measurements from the immediate solution resource usage log. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_TruncateResourceUsageLog (  
    @IdStart bigint  
);
```

@IdStart: The identifier of resource usage measurements at which removal of records will begin. The server MUST remove all resource usage measurements from the immediate solution resource usage log whose identifier is less than or equal to the specified value.

Return Code Values: An integer value which MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.4.79 **proc_TruncateResourceUsageWindowed**

The **proc_TruncateResourceUsageWindowed** stored procedure is called to remove resource usage measurements from the windowed solution resource usage log. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_TruncateResourceUsageWindowed (  
    @IdStart bigint  
);
```

@IdStart: The identifier of resource usage measurements at which removal of records will begin. The server MUST remove all resource usage measurements from the windowed solution resource usage log whose identifier is less than or equal to the specified value.

Return Code Values: An integer value which MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.4.80 **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,  
    @ListWebId              uniqueidentifier,  
    @ListId                 uniqueidentifier,  
    @ViewId                 uniqueidentifier,
```

```

@DisplayName      nvarchar(255),
@Type             tinyint,
@Flags           int,
@PageUrlID        uniqueidentifier,
@Level           tinyint,
@RequestGuid      uniqueidentifier = NULL OUTPUT
);

```

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

@ListWebId: The Site identifier for the Site that contains the list. 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 [2.2.3.14](#)) 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 containing the Web Part being updated. 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.

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

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.
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.

3.1.4.81 **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,
@Synchronization int,
@Type            int,
@SequenceNumber  int,
@Assembly        nvarchar(256),
@class           nvarchar(256),
@Data            nvarchar(256),
@Filter          nvarchar(256),
@Credential      int,
@RequestGuid     uniqueidentifier = NULL OUTPUT,
);

```

@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 0.

@Synchronization: Specifies the synchronicity of the event receiver and the action triggering the event. The value MUST be an integer which is listed in the following table:

Value	Name	Description
0	Default	For before event receivers, the server MUST run the event receiver synchronously. For after event receivers, the server is not required to run the event synchronously.
1	Synchronous	The server MUST run the event receiver using the same thread that is processing the request whose action triggered the event.
2	Asynchronous	The server MUST queue the task of running the event receiver. The server is not required to run the task using the same thread that is processing the request whose action triggered the event.

@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 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. 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 0.

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

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.

3.1.4.82 **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),
    @Synchronization         int,
    @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,
    @RequestGuid              uniqueidentifier = NULL OUTPUT
);
```

@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.

@Synchronization: Specifies the synchronicity of the event receiver and the action triggering the event. The value MUST be an integer which is listed in the following table:

Value	Name	Description
0	Default	For before event receivers, the server MUST run the event receiver synchronously. For after event receivers, the server is not required to run the event synchronously.
1	Synchronous	The server MUST run the event receiver using the same thread that is processing the request whose action triggered the event.
2	Asynchronous	The server MUST queue the task of running the event receiver. The server is not required to run the task using the same thread that is processing the request whose action triggered the event.

@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 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. 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.

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

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 Event Receiver type of the event receiver is identified by 32767 (as specified in [MS-WSSFO], section 2.2.3.6).

Result Sets: MUST NOT return any result sets.

3.1.4.83 **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,
    @ListWebId              uniqueidentifier,
    @ListId                 uniqueidentifier,
    @ViewId                 uniqueidentifier,
    @Flags                  int,
    @Type                   tinyint,
    @PageUrlID              uniqueidentifier,
    @Level                  tinyint,
    @RequestGuid            uniqueidentifier
);
```

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

@ListWebId: The Site identifier for the Site that contains the list. 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 containing the Web Part being updated. 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.

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

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.

3.1.4.84 **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           varbinary(max),
    @InstanceDataSize      int,
    @Modifications          nvarchar(max),
    @WakeupTime             datetime,
    @InstanceDataVersionId int,
    @Status1               int,
    @Status2               int,
    @Status3               int,
    @Status4               int,
    @Status5               int,
    @Status6               int,
    @Status7               int,
    @Status8               int,
    @Status9               int,
    @Status10              int,
    @ActivityDetails        varbinary(max),
    @WorkflowCompleted      bit,
```

```

        @WorkflowSuspended          bit,
        @WorkflowFaulting           bit,
        @WorkflowTerminated         bit,
        @WorkflowCanceled           bit,
        @UnlockInstance             bit,
        @ProcessingId               uniqueidentifier,
        @InternalState              int OUTPUT,
        @RequestGuid                uniqueidentifier = NULL OUTPUT
    );

```

@SiteId: The Site Collection identifier of the Site Collection which contains the Workflow. The protocol server MUST update the Site Collection Quota (section [3.1.1.5](#)) 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.

@ActivityDetails: An implementation defined binary payload containing the activity details.

@WorkflowCompleted: Determines whether the Workflow is marked as completed. Once a Workflow is marked as completed, it cannot be marked as not completed. 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 processing 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.

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

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 return one empty result set.

3.1.4.85 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,  
    @NewStatus             int = -1,  
    @RequestGuid           uniqueidentifier = NULL OUTPUT  
);
```

@SiteId: The Site Collection identifier of the Site Collection containing the Workflow. The server MUST update the Site Collection Quota (section [3.1.1.5](#)) 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 be either 0 or 1. 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 processing 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.

@NewStatus: The optional [Workflow Status1](#) value for the Workflow to be updated.

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

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.

3.1.4.86 **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                 nvarchar(max),
    @RequestGuid            uniqueidentifier = NULL OUTPUT
);
```

@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 [2.2.3.2](#)) or HTML format.

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

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

Value	Description
0	Successful completion.
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.

3.1.4.87 **proc_UpdateListViewToDataViewForSite**

The `proc_UpdateListViewToDataViewForSite` is called to change the Web Part Type of a set of List Web Parts to a different Web Part Type.

```

PROCEDURE proc_UpdateListViewToDataViewForSite (
    @DataViewId            uniqueidentifier,
    @FeatureId             uniqueidentifier,
    @ScopeId               uniqueidentifier,
    @ListViewId            uniqueidentifier
);

```

@DataViewId: The Web Part type identifier of the new Web Part Type. The server MUST NOT update the Web Part if the web part has CAML (that is, tp_View is not NULL), if the View Flags (as specified in [\[MS-WSSFO\]](#), section [2.2.2.11](#)) do not contain the value 1 (HTML view), or if the View Flags contain any bits in the mask 0x040A0810.

@FeatureId: The identifier of the feature that defines the Web Parts.

@ScopeId: The Site Collection identifier of the Site Collection containing the web parts.

@ListViewId: The Web Part type identifier of the web parts to update.

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

Result Sets: MUST NOT return any result sets.

3.1.4.88 proc_UpdateListViewToDataViewForWeb

The proc_UpdateListViewToDataViewForWeb is called to change the Web Part Type of a set of List Web Parts to a different Web Part Type.

```

PROCEDURE proc_UpdateListViewToDataViewForWeb (
    @DataViewId            uniqueidentifier,
    @FeatureId             uniqueidentifier,
    @ScopeId               uniqueidentifier,
    @ListViewId            uniqueidentifier
);

```

@DataViewId: The Web Part type identifier of the new Web Part Type. The server MUST NOT update the Web Part if the web part has CAML (that is, tp_View is not NULL), if the View Flags (as specified in [\[MS-WSSFO\]](#), section [2.2.2.11](#)) do not contain the value 1 (HTML view), or if the View Flags contain any bits in the mask 0x040A0810.

@FeatureId: The identifier of the feature that defines the Web Parts.

@ScopeId: The Site identifier for the Site that contains the web parts.

@ListViewId: The Web Part type identifier of the Web Parts to update.

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

Result Sets: MUST NOT return any result sets.

3.1.4.89 proc_UpdateSiteResourceUsage

The proc_UpdateSiteResourceUsage stored procedure is called to update resource usage values for a site collection. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_UpdateSiteResourceUsage (

```

```

        @SiteId                uniqueidentifier,
        @CurrentResourceUsage   float,
        @AverageResourceUsage   float
    );

```

@SiteId: The site collection identifier of the site collection for which resource usage values are being updated.

@CurrentResourceUsage: The resource usage value for the current monitoring interval.

@AverageResourceUsage: The mean resource usage value for the specified site collection over the configured retention interval for resource usage data.

Return Code Values: An integer which MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.4.90 proc_UpdateSolution

The proc_UpdateSolution stored procedure is called to update information about a sandboxed solution. The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_UpdateSolution (
    @SiteId                uniqueidentifier,
    @SolutionId            uniqueidentifier,
    @Hash                  nvarchar(50),
    @ValidatorsHash        char(64),
    @ValidationErrorUrl    nvarchar(1024),
    @ValidationErrorMessage nvarchar(1024)
);

```

@SiteId: The site collection identifier of the site collection for the sandboxed solution This value MUST NOT be NULL.

@SolutionId: The identifier of the sandboxed solution. This value MUST NOT be NULL.

@Hash: The implementation-specific hash of the content of the sandboxed solution. This value MUST NOT be NULL.

@ValidatorsHash: The implementation-specific hash of the validators that validated the sandboxed solution. This value MUST NOT be NULL.

@ValidationErrorUrl: If validation of the sandboxed solution failed, MUST contain the URL with more information about the validation failure.

@ValidationErrorMessage: If validation of the sandboxed solution failed, MUST contain the specific error message of the validation failure.

Return Code Values:

Value	Description
0	Successful execution.
1	The sandboxed solution cannot be found.

Result Sets: MUST NOT return any result sets.

3.1.4.91 **proc_UpdateSolutionResourceUsage**

The `proc_UpdateSolutionResourceUsage` stored procedure is called to update resource usage values for a sandboxed solution. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_UpdateSolutionResourceUsage (
    @SiteId                uniqueidentifier,
    @SolutionId            uniqueidentifier,
    @ResourceQuota          float,
    @RecentInvocations      int
);
```

@SiteId: The site collection identifier of the site collection containing the sandboxed solution.

@SolutionId: The identifier of the sandboxed solution.

@ResourceQuota: The resource usage value for the sandboxed solution over the current monitoring interval.

@RecentInvocations: The count of invocations of code from this sandboxed solution over the current monitoring interval.

Return Code Values: An integer value which MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.4.92 **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,
    @ListWebId              uniqueidentifier,
    @ListId                uniqueidentifier,
    @ViewId                uniqueidentifier,
    @View                  tCompressedString,
    @DisplayName            nvarchar(255),
    @ContentTypeId          varbinary(512),
    @Type                  tinyint,
    @Flags                  int,
    @BaseViewID             tinyint,
    @PageUrlID             uniqueidentifier,
    @Level                  tinyint,
    @RequestGuid            uniqueidentifier = NULL OUTPUT
);
```

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

@ListWebId: The Site identifier for the Site that contains the list. 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 @DisplayName 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 protocol server MUST update the view flags property to the value of @Flags, and 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.

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

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.

3.1.4.93 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               varbinary(512),
    @WebPartID              uniqueidentifier,
    @WebPartTypeID          uniqueidentifier,
    @Assembly               nvarchar(255),
    @Class                  nvarchar(255),
    @SolutionId             uniqueidentifier,
    @bCheckLock             bit,
    @IsIncluded             bit,
    @FrameState             tinyint,
    @ZoneID                 nvarchar(64),
    @PartOrder              int,
    @TheFlags               int,
    @TheType                tinyint,
    @TheBaseViewID          tinyint,
    @AllUsersProperties      varbinary(max),
    @PerUserProperties       varbinary(max),
    @WebPartIdProperty      nvarchar(255),
    @RequestGuid            uniqueidentifier = NULL OUTPUT
);
```

@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 SystemID of 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.

@Assembly: The fully qualified name of the assembly that implements the web part.

@Class: The name of the .NET class that implements the web part.

@SolutionId: The identifier of the sandboxed solution that implements the web part.

@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.

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

@TheType: The page type of the web part page containing the web part.

@TheBaseViewID: The base view identifier of the web part.

@AllUsersProperties: A binary payload containing 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 binary payload containing 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.

@WebPartIdProperty: The optional identifier of the Web Part. If not NULL, it MUST be unique per page, not begin with a number, and contain only a combination of alphanumeric characters and underscores.

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

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.

Value	Description
87	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 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.

3.1.4.94 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               varbinary(512),
    @WebPartID              uniqueidentifier,
    @Cache                  varbinary(max),
    @RequestGuid            uniqueidentifier = NULL OUTPUT
);

```

@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.

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

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.

3.1.4.95 **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,
    @RequestGuid            uniqueidentifier = NULL OUTPUT
);

```

@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.

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

Return Code Values: The stored procedure MUST return an integer listed in the following table:

Value	Description
0	Successful completion.
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 @UserId is NULL.
212	The Site Collection is Locked.
1816	The Quota for the Site Collection has been exceeded.
- 2147467259	An error occurred while the stored procedure was running.

Result Sets: MUST NOT return any result sets.

3.1.4.96 **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,
    @Type                   tinyint,
    @Flags                  int,
    @IsIncluded             bit,
    @FrameState             tinyint,
    @AllUsersproperties     varbinary(max),
    @PerUserproperties      varbinary(max),
    @WebPartIdProperty      nvarchar(255),
    @Level                  tinyint OUTPUT,
    @RequestGuid            uniqueidentifier = NULL 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.

@Type: The Page Type, as specified in [\[MS-WSSFO\]](#), section [2.2.3.14](#), of the Web Part Page that contains the Web Part.

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

@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.

@WebPartIdProperty: The optional identifier of the Web Part. If not NULL, it MUST be unique per page, not begin with a number, and contain only a combination of alphanumeric characters and underscores.

@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.

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

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

Value	Description
0	Successful completion.
212	The Site Collection is Locked.
1816	The Quota for the Site Collection has been exceeded.
-2147467259	An error occurred while the stored procedure was running.

Result Sets: MUST NOT return any result sets.

3.1.4.97 **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,  
    @Assembly               nvarchar(255),  
    @Class                  nvarchar(255),  
    @SolutionId             uniqueidentifier,  
    @RequestGuid            uniqueidentifier = NULL OUTPUT  
);
```

@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.

@Assembly: The fully qualified name of the assembly that implements the web part.

@Class: The name of the .NET class that implements the web part.

@SolutionId: The identifier of the sandboxed solution that implements the web part.

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

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.

3.1.4.98 **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,  
    @Assembly               nvarchar(255),  
    @RequestGuid            uniqueidentifier = NULL OUTPUT  
);
```

```

@Class                nvarchar(255),
@SolutionId           uniqueidentifier,
@TheListID            uniqueidentifier,
@IsIncluded           bit,
@FrameState           tinyint
@ZoneID               nvarchar(64),
@PartOrder            int,
@TheFlags             int,
@TheType              tinyint,
@TheBaseViewID        tinyint,
@ContentTypeID        varbinary(512),
@Source               nvarchar(max),
@AllUsersProperties   varbinary(max),
@PerUserProperties     varbinary(max),
@WebPartIdProperty    nvarchar(255),
@RequestGuid          uniqueidentifier = NULL OUTPUT
);

```

@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 [2.2.1.15](#)) of the Web Part within the Site Collection. If @WebPartID matches the Web Part identifier (as specified in [\[MS-WSSFO\]](#), section [2.2.1.15](#)) of an existing Web Part on the different Web Part Page, then the protocol server MUST generate a new Web Part identifier (as specified in [\[MS-WSSFO\]](#), section [2.2.1.15](#)). 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.

@Assembly: The fully qualified name of the assembly that implements the web part.

@Class: The name of the .NET class that implements the web part.

@SolutionId: The identifier of the sandboxed solution that implements the web part.

@TheListID: The list identifier of the list associated with the web part.

@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.

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

@TheType: The web part type identifier of the web part.

@TheBaseViewID: The base view identifier 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 [2.2.3.2](#)) or HTML format.

@AllUsersProperties: A binary payload containing 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 binary payload containing 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.

@WebPartIdProperty: The optional identifier of the Web Part. If not NULL, it MUST be unique per page, not begin with a number, and contain only a combination of alphanumeric characters and underscores.

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

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.
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.

3.1.4.99 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     nvarchar(max),
@Version                int,
@RequestGuid            uniqueidentifier = NULL OUTPUT
);

```

@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 rights mask (as specified in [\[MS-WSSFO\]](#), section [2.2.2.13](#)) 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.

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

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.

3.1.4.100 `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         varbinary(max),  
    @TextPayload           nvarchar(max),  
    @ProcessingId          uniqueidentifier,  
    @ForceUpdate           bit= 0,  
    @RequestGuid           uniqueidentifier = NULL OUTPUT  
);
```

@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 Processing 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 Processing 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 Processing 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.

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

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.

3.1.4.101 **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,  
    @RequestGuid            uniqueidentifier = NULL OUTPUT  
);
```

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

@WorkflowInstanceId: The Workflow identifier of the Workflow.

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

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 the front-end Web server for all back end database server connections.

3.2.3 Initialization

The front-end Web server MUST validate the user making the request before calling the stored procedures. The Site Collection identifier (as specified in [\[MS-WSSFO\]](#), section [2.2.1.9](#)) and the User identifier (as specified in [\[MS-WSSFO\]](#), section [2.2.1.13](#)) for the user making the request are looked up by the front-end Web server before calling additional stored procedures.

3.2.4 Message Processing Events and Sequencing Rules

The front-end Web server handles each stored procedure with the same processing method of calling the stored procedure and waiting for the 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.

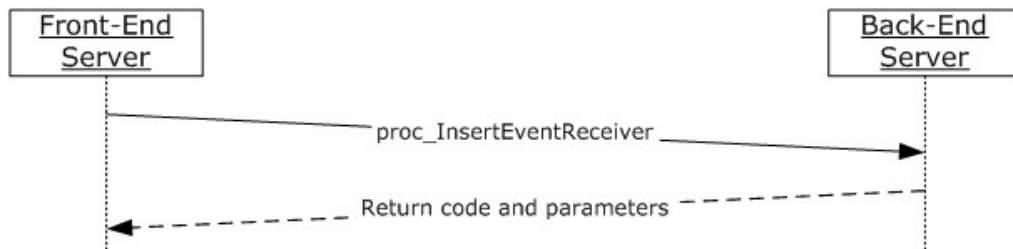


Figure 2: Create an event receiver

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 the [proc_InsertEventReceiver](#) stored procedure to save the event receiver registration to back end database server.
3. The `proc_InsertEventReceiver` stored procedure 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 [\[MS-WSSFO\]](#), section [4.5](#), 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.

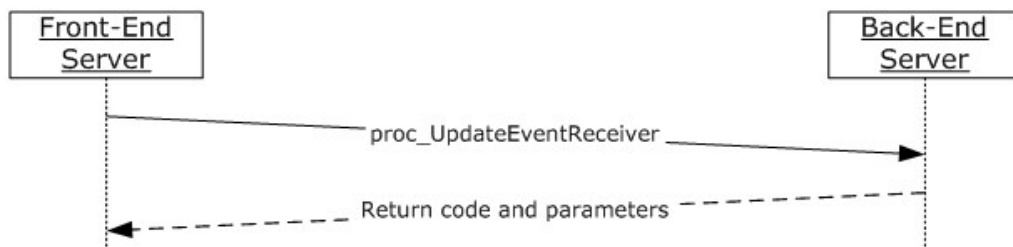


Figure 3: Update an event receiver

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 the [proc_UpdateEventReceiver](#) stored procedure to save the event receiver's new properties to the back end database server.
3. The `proc_UpdateEventReceiver` stored procedure 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.

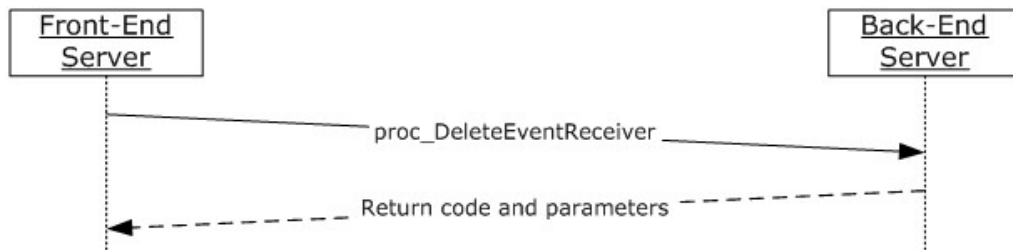


Figure 4: Delete an event receiver

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 the [proc_DeleteEventReceiver](#) stored procedure to delete the event receiver in the back end database server.
3. The `proc_DeleteEventReceiver` stored procedure 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.

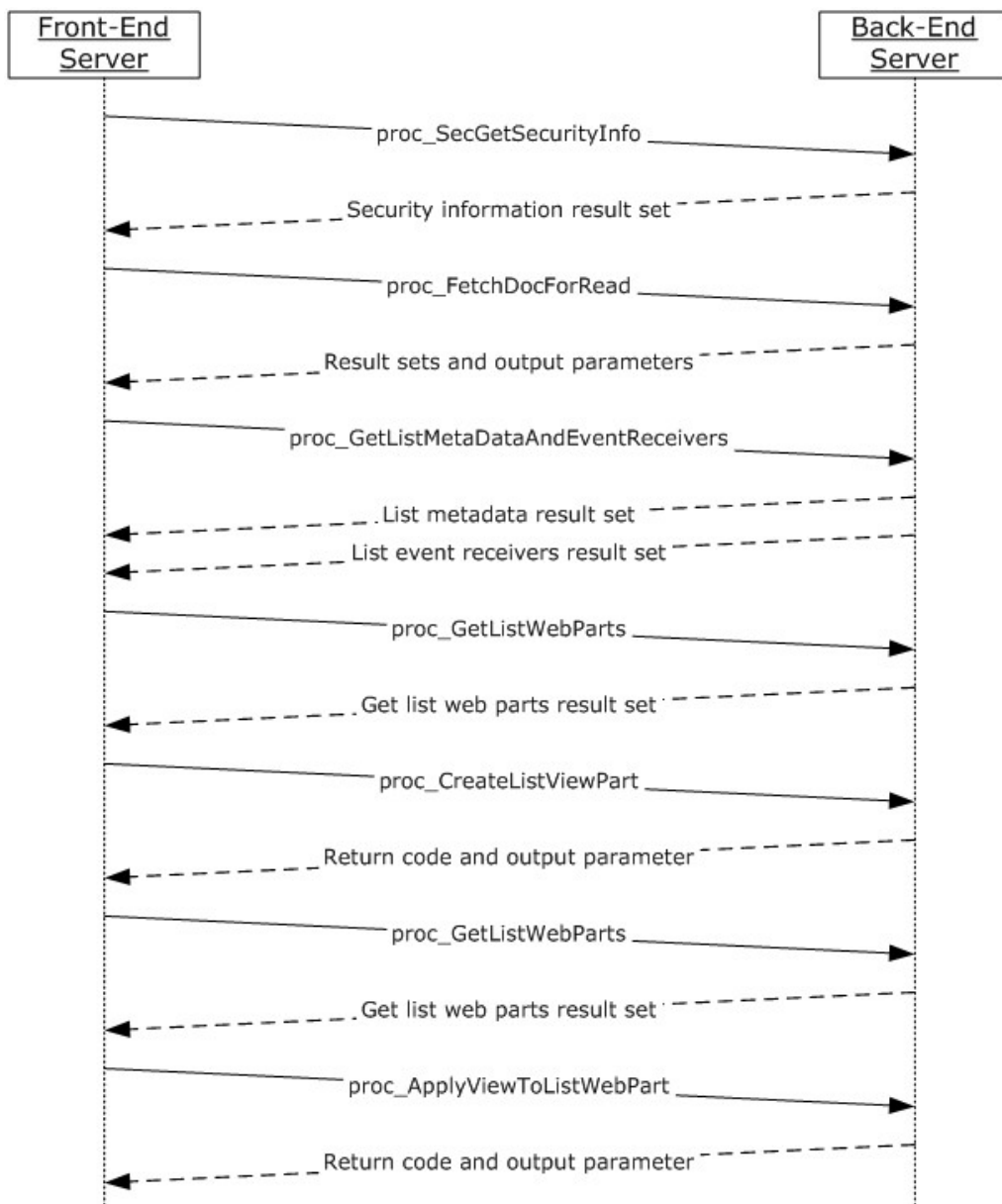


Figure 5: Add a List View Web Part

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:

The front-end Web server retrieves security permissions information about the requested Site. It does this by calling the [\[MS-WSSFO\].proc_SecGetSecurityInfo](#) stored procedure.

The back-end database server returns the Security Information Result Set, which consists of information about security permissions about the requested Site.

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-WSSFO].proc_FetchDocForRead stored procedure.

The Back-End Database Server returns a set of Result Sets detailed in [MS-WSSFO], sections 3.1.4.14.1–3.1.4.14.3, and the Publishing Level of the Document as an output parameter.

The front-end Web server then fetches properties of the list that the list View Web Part refers to by calling the [MS-WSSFO].proc_GetListMetaDataAndEventReceivers stored procedure.

The Back-End Database Server returns two Result Sets which include the metadata and Event Receivers for the specified list.

The front-end Web server then fetches Views associated with the list by calling the [proc_GetListWebParts](#) stored procedure.

The Back-End Database Server returns one Result Set which include the list views associated with the list.

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 places the list View Web Part in the specified Web Part Zone on the specified Web Part Page by calling the [proc_CreateListViewPart](#) stored procedure.

The Back-End Database Server returns an output code and the Publishing Level as the output parameter.

The front-end Web server then re-fetches the Views corresponding to the list by calling the [proc_GetListWebParts](#) stored procedure.

The Back-End Database Server returns one Result Set which include the list Views corresponding to the list.

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.

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.

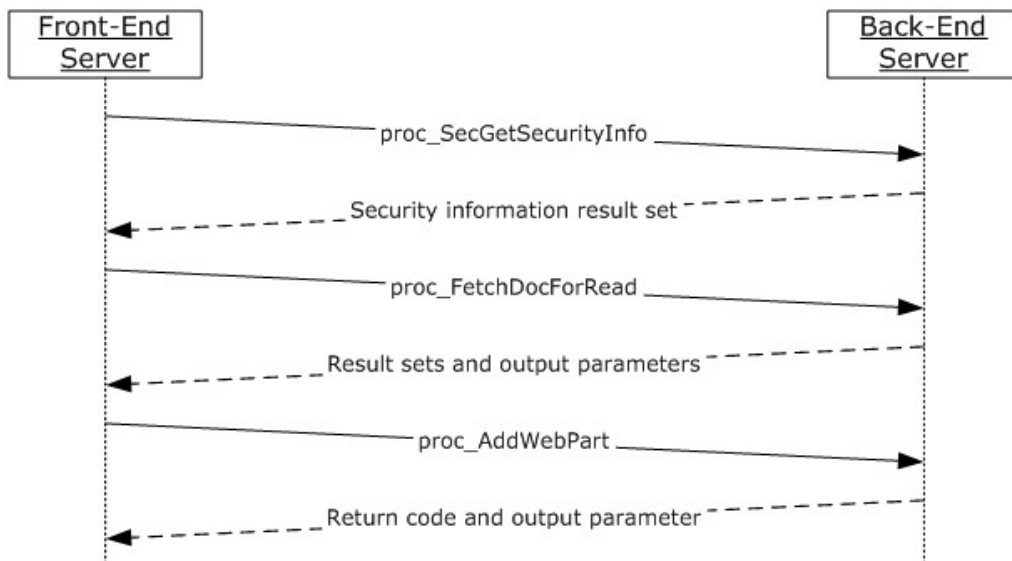


Figure 6: Add a non-List View Web Part

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 [\[MS-WSSFO\].proc_SecGetSecurityInfo](#) stored procedure.
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 Web Part needs to be added by calling the [\[MS-WSSFO\].proc_FetchDocForRead](#) stored procedure.
4. The Back-End Database Server returns a set of Result Sets detailed in [\[MS-WSSFO\], sections 3.1.4.14.1–3.1.4.14.13](#) 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.

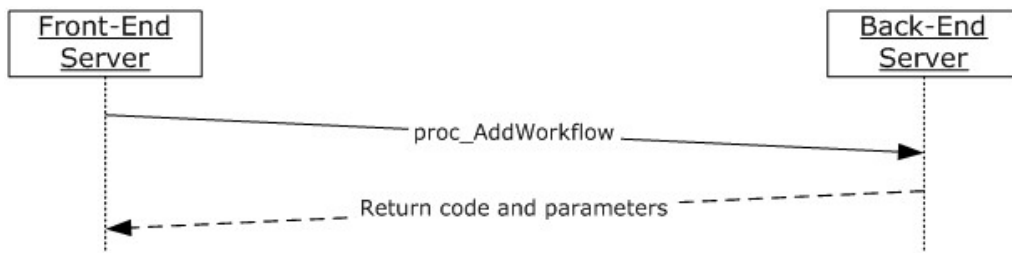


Figure 9: Create a Workflow for a List Item

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.

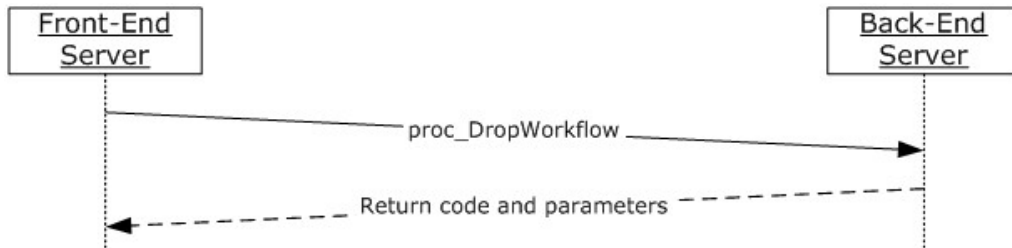


Figure 10: Delete a Workflow from a List Item

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.

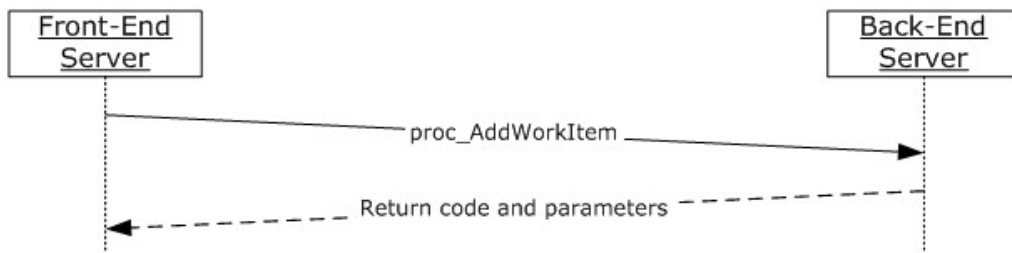


Figure 11: Create a Work Item for Bulk Editing Workflow Tasks

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.

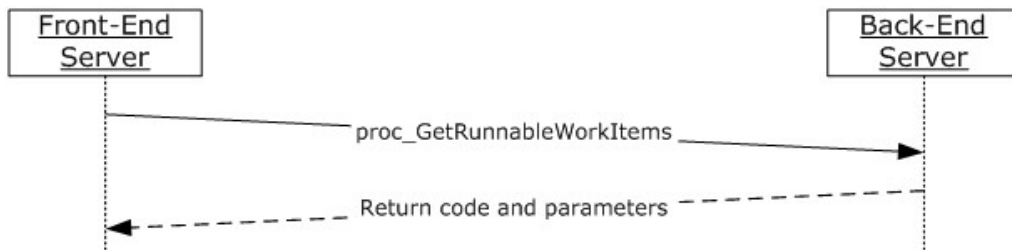


Figure 12: Retrieve a Set of Runnable Bulk Workflow Task Work Items

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 completed execution of a Work Item that have delivery dates that have passed and is about to mark them as completed.

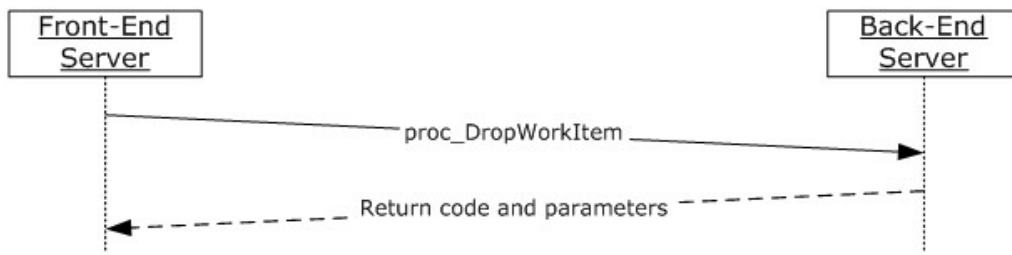


Figure 13: Delete a Work Item

The following actions happen:

1. The front-end Web server requests to mark Work Items as completed 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. Refer to [\[MS-SPPTC\]](#) for security considerations relating to sandboxed solutions.

5.2 Index of Security Parameters

None.

6 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- Microsoft® SharePoint® Foundation 2010
- Microsoft® SQL Server® 2005
- Microsoft® SQL Server® 2008
- Microsoft® SQL Server® 2008 R2

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

[<1> Section 2.2.6.6:](#) 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 3.1.4.7:](#) 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

A

Abstract data model

[client](#) 138

[server](#) 28

[Add a list view Web Part example](#) 140

[Add a non-list view Web Part example](#) 142

[Applicability](#) 13

[Attribute groups - overview](#) 27

[Attributes - overview](#) 27

B

[Binary structures - overview](#) 19

Bit fields

[event receiver source type bit field](#) 16

[workflow association configuration](#) 17

[workflow internal state](#) 18

[Workflow Status1](#) 18

C

[Capability negotiation](#) 13

[Change tracking](#) 150

Client

[abstract data model](#) 138

[Content Database Programmability Extensions](#)

[Communications interface](#) 138

[initialization](#) 138

[local events](#) 138

[message processing](#) 138

[overview](#) 138

[sequencing rules](#) 138

[timer events](#) 138

[timers](#) 138

Common data types

[overview](#) 16

[Complex types - overview](#) 26

[Content Database Programmability Extensions](#)

[Communications interface](#) 138

[Context collection identifier simple type](#) 16

[Context identifier simple type](#) 16

[Context object identifier simple type](#) 16

[Context type identifier simple type](#) 16

[Create a work item for bulk editing workflow tasks example](#) 145

[Create a workflow for a list item example](#) 144

[Create an event receiver example](#) 139

D

Data model - abstract

[client](#) 138

[server](#) 28

Data types

[common](#) 16

[context collection identifier simple type](#) 16

[context identifier simple type](#) 16

[context object identifier simple type](#) 16

[context type identifier simple type](#) 16

[event receiver source identifier simple type](#) 16

[list item version simple type](#) 16

[workflow template identifier simple type](#) 16

Data types - simple

[context identifier](#) 16

[context object identifier](#) 16

[context type identifier](#) 16

[event receiver source identifier](#) 16

[ontext collection identifier](#) 16

[workflow template identifier](#) 16

Data types = simple

[list item version](#) 16

[Delete a Web Part example](#) 144

[Delete a work item example](#) 146

[Delete a workflow from a list item example](#) 145

[Delete an event receiver example](#) 140

E

Elements

[Workflow Modifications](#) 26

[Elements - overview](#) 26

Enumerations

[sandboxed solution status](#) 19

[Event operations overview](#) 12

[Event receiver source identifier simple type](#) 16

[Event receiver source type bit field](#) 16

Events

[local - client](#) 138

[local - server](#) 137

[timer - client](#) 138

[timer - server](#) 137

Examples

[add a list view Web Part](#) 140

[add a non-list view Web Part](#) 142

[create a work item for bulk editing workflow tasks](#) 145

[create a workflow for a list item](#) 144

[create an event receiver](#) 139

[delete a Web Part](#) 144

[delete a work item](#) 146

[delete a workflow from a list item](#) 145

[delete an event receiver](#) 140

[get all Web Parts on a Web Part page](#) 143

[overview](#) 139

[read event receivers](#) 139

[retrieve a set of runnable bulk workflow task work items](#) 146

[update an event receiver example](#) 139

F

[Fields - vendor-extensible](#) 13

G

[Get all Web Parts on a Web Part page example](#) 143

[Glossary](#) 8

[Groups - overview](#) 27

I

[Implementer - security considerations](#) 148

[Index of security parameters](#) 148

[Informative references](#) 11

Initialization

[client](#) 138

[server](#) 31

Interfaces – client

[Content Database Programmability Extensions](#)

[Communications](#) 138

[Introduction](#) 8

L

[List item version simple type](#) 16

[List Item Workflows result set](#) 19

Local events

[client](#) 138

[server](#) 137

M

Message processing

[client](#) 138

[server](#) 31

Messages

[attribute groups](#) 27

[attributes](#) 27

[binary structures](#) 19

[common data types](#) 16

[complex types](#) 26

[elements](#) 26

[event receiver source type bit field](#) 16

[groups](#) 27

[List Item Workflows result set](#) 19

[namespaces](#) 26

[simple types](#) 26

[Solution Resource Usage Processing result set](#) 21

[Solution Resource Usage result set](#) 21

[table structures](#) 26

[transport](#) 16

[view structures](#) 26

[Web Parts result set](#) 22

[Work Items result set](#) 25

[workflow association configuration bit field](#) 17

[Workflow Associations result set](#) 23

[workflow internal state bit field](#) 18

[Workflow Modifications element](#) 26

[Workflow Status1 bit field](#) 18

[XML structures](#) 26

Methods

[proc_AddNonListViewFormPersonalization](#) 31

[proc_AddNonListViewFormWebPartForUrl](#) 33

[proc_AddSolution](#) 35

[proc_AddWebPart](#) 36

[proc_AddWorkflow](#) 38

[proc_AddWorkflowAssociation](#) 40

[proc_AddWorkItem](#) 41

[proc_ApplyViewToListWebPart](#) 43

[proc_AutoCleanupWorkflows](#) 45

[proc_AutoDropWorkflows](#) 45

[proc_CancelDeclarativeWorkflows](#) 47

[proc_CancelWorkflow](#) 47

[proc_CommitUpdatedZoneIds](#) 47

[proc_CompleteInProgressWorkItems](#) 48

[proc_CopyDefaultViewWebParts](#) 49

[proc_CountWorkflowAssociations](#) 50

[proc_CountWorkflows](#) 50

[proc_CountWorkflowsBatch](#) 51

[proc_CreateListViewPart](#) 52

[proc_DeleteDocEventReceiver](#) 54

[proc_DeleteEventReceiver](#) 55

[proc_DeleteEventReceiversBySourceId](#) 57

[proc_DeleteInProgressWorkItems](#) 58

[proc_DeleteSmartPagePersonalization](#) 59

[proc_DeleteWebPart](#) 59

[proc_DeleteWebPartPersonalization](#) 61

[proc_DeleteWebPartWhileSaving](#) 62

[proc_DeleteZoneWebPartsWhileSaving](#) 62

[proc_DisableAssociationsForTemplate](#) 63

[proc_DropWorkflow](#) 64

[proc_DropWorkflowAssociation](#) 64

[proc_DropWorkItem](#) 65

[proc_EnableDeclarativeWorkflowAssociations](#) 65

[proc_EnumerateWebPartsForList](#) 66

[proc_EnumerateWebPartsForWeb](#) 66

[proc_EnumResourceWarningSites](#) 68

[proc_FailOverInProgressWorkItems](#) 69

[proc_GetAllResourceUsageForSiteToday](#) 69

[proc_GetAllWebPartsOnPage](#) 70

[proc_GetAverageDailyResourceUsageForSite](#) 71

[proc_GetContextCollectionEventReceivers](#) 72

[proc_GetContextObjectEventReceivers](#) 73

[proc_GetDocEventReceivers](#) 74

[proc_GetListItemWorkflows](#) 74

[proc_GetListItemWorkflowWithInstanceDataAndLock](#) 76

[proc_GetListWebParts](#) 77

[proc_GetNextWebPartOrder](#) 79

[proc_GetRecycleBinItemEventReceivers](#) 79

[proc_GetRunnableWorkItems](#) 81

[proc_GetRunningWorkBatchCount](#) 82

[proc_GetSiteResourceUsage](#) 83

[proc_GetSiteSolutionResourceUsage](#) 83

[proc_GetSolutionInfo](#) 84

[proc_GetSolutionResourceQuota](#) 85

[proc_GetSolutionResourceUsage](#) 85

[proc_GetSolutionResourceUsageDailyOrdinal](#) 86

[proc_GetSolutionsData](#) 86

[proc_GetWFTemplatesLastModifiedForWeb](#) 87

[proc_GetWorkflowAssociations](#) 88

[proc_GetWorkflowDataForItem](#) 89

[proc_GetWorkItems](#) 90

[proc_InsertContextEventReceiver](#) 91

[proc_InsertDocEventReceiver](#) 93

[proc_InsertEventReceiver](#) 95

[proc_LogSolutionResourceUsage20](#) 97

[proc_LogSolutionResourceUsageDaily20](#) 99

[proc_LogSolutionResourceUsageWindowed20](#) 102

[proc_ProcessSolutionResourceUsageLogData](#) 105

proc ProcessSolutionResourceUsageWindowedData	105	proc CancelWorkflow method	47
proc ProvisionWebPart	106	proc CommitUpdatedZoneIds method	47
proc RemoveSolution	107	proc CompleteInProgressWorkItems method	48
proc ResetSiteResourceUsageWarnings	107	proc CopyDefaultViewWebParts method	49
proc RestoreWebPartForDoc	108	proc CountWorkflowAssociations method	50
proc RevertInProgressWorkItem	108	proc CountWorkflows method	50
proc RevertInProgressWorkItems	109	proc CountWorkflowsBatch method	51
proc SetEventReceiverToSynchronous	110	proc CreateListViewPart method	52
proc TruncateResourceUsageDaily	110	proc DeleteDocEventReceiver method	54
proc TruncateResourceUsageLog	111	proc DeleteEventReceiver method	55
proc TruncateResourceUsageWindowed	111	proc DeleteEventReceiversBySourceId method	57
proc UpdateDataViewWhileSaving	111	proc DeleteInProgressWorkItems method	58
proc UpdateDocEventReceiver	112	proc DeleteSmartPagePersonalization method	59
proc UpdateEventReceiver	114	proc DeleteWebPart method	59
proc UpdateListFormWhileSaving	116	proc DeleteWebPartPersonalization method	61
proc UpdateListItemWorkflowInstanceData	117	proc DeleteWebPartWhileSaving method	62
proc UpdateListItemWorkflowLock	120	proc DeleteZoneWebPartsWhileSaving method	62
proc UpdateListViewFormWebPartSource	121	proc DisableAssociationsForTemplate method	63
proc UpdateListViewToDataViewForSite	121	proc DropWorkflow method	64
proc UpdateListViewToDataViewForWeb	122	proc DropWorkflowAssociation method	64
proc UpdateSiteResourceUsage	122	proc DropWorkItem method	65
proc UpdateSolution	123	proc EnableDeclarativeWorkflowAssociations method	65
proc UpdateSolutionResourceUsage	124	proc EnumerateWebPartsForList method	66
proc UpdateViewWhileSaving	124	proc EnumerateWebPartsForWeb method	66
proc UpdateWebPart	126	proc EnumResourceWarningSites method	68
proc UpdateWebPartCache	128	proc FailOverInProgressWorkItems method	69
proc UpdateWebPartIsIncluded	129	proc GetAllResourceUsageForSiteToday method	69
proc UpdateWebPartProps	130	proc GetAllWebPartsOnPage method	70
proc UpdateWebPartTypeId	132	proc GetAverageDailyResourceUsageForSite method	71
proc UpdateWebPartWhileSaving	132	proc GetContextCollectionEventReceivers method	72
proc UpdateWorkflowAssociation	134	proc GetContextObjectEventReceivers method	73
proc UpdateWorkItem	136	proc GetDocEventReceivers method	74
proc WorkflowHasVisibleParentItem	137	proc GetListItemWorkflows method	74
		proc GetListItemWorkflowWithInstanceDataAndLock method	76
N		proc GetListWebParts method	77
Namespaces	26	proc GetNextWebPartOrder method	79
Normative references	11	proc GetRecycleBinItemEventReceivers method	79
O		proc GetRunnableWorkItems method	81
Overview (synopsis)	11	proc GetRunningWorkBatchCount method	82
P		proc GetSiteResourceUsage method	83
Parameters - security index	148	proc GetSiteSolutionResourceUsage method	83
Preconditions	13	proc GetSolutionInfo method	84
Prerequisites	13	proc GetSolutionResourceQuota method	85
proc AddNonListViewFormPersonalization method	31	proc GetSolutionResourceUsage method	85
proc AddNonListViewFormWebPartForUrl method	33	proc GetSolutionResourceUsageDailyOrdinal method	86
proc AddSolution method	35	proc GetSolutionsData method	86
proc AddWebPart method	36	proc GetWFTemplatesLastModifiedForWeb method	87
proc AddWorkflow method	38	proc GetWorkflowAssociations method	88
proc AddWorkflowAssociation method	40	proc GetWorkflowDataForItem method	89
proc AddWorkItem method	41	proc GetWorkItems method	90
proc ApplyViewToListWebPart method	43	proc InsertContextEventReceiver method	91
proc AutoCleanupWorkflows method	45	proc InsertDocEventReceiver method	93
proc AutoDropWorkflows method	45	proc InsertEventReceiver method	95
proc CancelDeclarativeWorkflows method	47	proc LogSolutionResourceUsage20 method	97
		proc LogSolutionResourceUsageDaily20 method	99

[proc_LogSolutionResourceUsageWindowed20 method](#) 102
[proc_ProcessSolutionResourceUsageLogData method](#) 105
[proc_ProcessSolutionResourceUsageWindowedData method](#) 105
[proc_ProvisionWebPart method](#) 106
[proc_RemoveSolution method](#) 107
[proc_ResetSiteResourceUsageWarnings method](#) 107
[proc_RestoreWebPartForDoc method](#) 108
[proc_RevertInProgressWorkItem method](#) 108
[proc_RevertInProgressWorkItems method](#) 109
[proc_SetEventReceiverToSynchronous method](#) 110
[proc_TruncateResourceUsageDaily method](#) 110
[proc_TruncateResourceUsageLog method](#) 111
[proc_TruncateResourceUsageWindowed method](#) 111
[proc_UpdateDataViewWhileSaving method](#) 111
[proc_UpdateDocEventReceiver method](#) 112
[proc_UpdateEventReceiver method](#) 114
[proc_UpdateListFormWhileSaving method](#) 116
[proc_UpdateListItemWorkflowInstanceData method](#) 117
[proc_UpdateListItemWorkflowLock method](#) 120
[proc_UpdateListViewFormWebPartSource method](#) 121
[proc_UpdateListViewToDataViewForSite method](#) 121
[proc_UpdateListViewToDataViewForWeb method](#) 122
[proc_UpdateSiteResourceUsage method](#) 122
[proc_UpdateSolution method](#) 123
[proc_UpdateSolutionResourceUsage method](#) 124
[proc_UpdateViewWhileSaving method](#) 124
[proc_UpdateWebPart method](#) 126
[proc_UpdateWebPartCache method](#) 128
[proc_UpdateWebPartIsIncluded method](#) 129
[proc_UpdateWebPartProps method](#) 130
[proc_UpdateWebPartTypeId method](#) 132
[proc_UpdateWebPartWhileSaving method](#) 132
[proc_UpdateWorkflowAssociation method](#) 134
[proc_UpdateWorkItem method](#) 136
[proc_WorkflowHasVisibleParentItem method](#) 137
[Product behavior](#) 149

R

[Read event receivers example](#) 139

References

[informative](#) 11
[normative](#) 11

[Relationship to other protocols](#) 12

Result sets - messages

[List Item Workflows](#) 19
[Solution Resource Usage](#) 21
[Solution Resource Usage Processing](#) 21
[Web Parts](#) 22
[Work Items](#) 25
[Workflow Associations](#) 23

[Retrieve a set of runnable bulk workflow task work items example](#) 146

S

[Sandboxed solution status enumeration](#) 19

Security

[implementer considerations](#) 148
[parameter index](#) 148

Sequencing rules

[client](#) 138
[server](#) 31

Server

[abstract data model](#) 28
[initialization](#) 31
[local events](#) 137
[message processing](#) 31
[proc_AddNonListViewFormPersonalization method](#) 31
[proc_AddNonListViewFormWebPartForUrl method](#) 33
[proc_AddSolution method](#) 35
[proc_AddWebPart method](#) 36
[proc_AddWorkflow method](#) 38
[proc_AddWorkflowAssociation method](#) 40
[proc_AddWorkItem method](#) 41
[proc_ApplyViewToListWebPart method](#) 43
[proc_AutoCleanupWorkflows method](#) 45
[proc_AutoDropWorkflows method](#) 45
[proc_CancelDeclarativeWorkflows method](#) 47
[proc_CancelWorkflow method](#) 47
[proc_CommitUpdatedZoneIds method](#) 47
[proc_CompleteInProgressWorkItems method](#) 48
[proc_CopyDefaultViewWebParts method](#) 49
[proc_CountWorkflowAssociations method](#) 50
[proc_CountWorkflows method](#) 50
[proc_CountWorkflowsBatch method](#) 51
[proc_CreateListViewPart method](#) 52
[proc_DeleteDocEventReceiver method](#) 54
[proc_DeleteEventReceiver method](#) 55
[proc_DeleteEventReceiversBySourceId method](#) 57
[proc_DeleteInProgressWorkItems method](#) 58
[proc_DeleteSmartPagePersonalization method](#) 59
[proc_DeleteWebPart method](#) 59
[proc_DeleteWebPartPersonalization method](#) 61
[proc_DeleteWebPartWhileSaving method](#) 62
[proc_DeleteZoneWebPartsWhileSaving method](#) 62
[proc_DisableAssociationsForTemplate method](#) 63
[proc_DropWorkflow method](#) 64
[proc_DropWorkflowAssociation method](#) 64
[proc_DropWorkItem method](#) 65
[proc_EnableDeclarativeWorkflowAssociations method](#) 65
[proc_EnumerateWebPartsForList method](#) 66
[proc_EnumerateWebPartsForWeb method](#) 66
[proc_EnumResourceWarningSites method](#) 68
[proc_FailOverInProgressWorkItems method](#) 69
[proc_GetAllResourceUsageForSiteToday method](#) 69
[proc_GetAllWebPartsOnPage method](#) 70
[proc_GetAverageDailyResourceUsageForSite method](#) 71

[proc_GetContextCollectionEventReceivers method](#) 72
[proc_GetContextObjectEventReceivers method](#) 73
[proc_GetDocEventReceivers method](#) 74
[proc_GetListItemWorkflows method](#) 74
[proc_GetListItemWorkflowWithInstanceDataAndLock method](#) 76
[proc_GetListWebParts method](#) 77
[proc_GetNextWebPartOrder method](#) 79
[proc_GetRecycleBinItemEventReceivers method](#) 79
[proc_GetRunnableWorkItems method](#) 81
[proc_GetRunningWorkBatchCount method](#) 82
[proc_GetSiteResourceUsage method](#) 83
[proc_GetSiteSolutionResourceUsage method](#) 83
[proc_GetSolutionInfo method](#) 84
[proc_GetSolutionResourceQuota method](#) 85
[proc_GetSolutionResourceUsage method](#) 85
[proc_GetSolutionResourceUsageDailyOrdinal method](#) 86
[proc_GetSolutionsData method](#) 86
[proc_GetWFTemplatesLastModifiedForWeb method](#) 87
[proc_GetWorkflowAssociations method](#) 88
[proc_GetWorkflowDataForItem method](#) 89
[proc_GetWorkItems method](#) 90
[proc_InsertContextEventReceiver method](#) 91
[proc_InsertDocEventReceiver method](#) 93
[proc_InsertEventReceiver method](#) 95
[proc_LogSolutionResourceUsage20 method](#) 97
[proc_LogSolutionResourceUsageDaily20 method](#) 99
[proc_LogSolutionResourceUsageWindowed20 method](#) 102
[proc_ProcessSolutionResourceUsageLogData method](#) 105
[proc_ProcessSolutionResourceUsageWindowedData method](#) 105
[proc_ProvisionWebPart method](#) 106
[proc_RemoveSolution method](#) 107
[proc_ResetSiteResourceUsageWarnings method](#) 107
[proc_RestoreWebPartForDoc method](#) 108
[proc_RevertInProgressWorkItem method](#) 108
[proc_RevertInProgressWorkItems method](#) 109
[proc_SetEventReceiverToSynchronous method](#) 110
[proc_TruncateResourceUsageDaily method](#) 110
[proc_TruncateResourceUsageLog method](#) 111
[proc_TruncateResourceUsageWindowed method](#) 111
[proc_UpdateDataViewWhileSaving method](#) 111
[proc_UpdateDocEventReceiver method](#) 112
[proc_UpdateEventReceiver method](#) 114
[proc_UpdateListFormWhileSaving method](#) 116
[proc_UpdateListItemWorkflowInstanceData method](#) 117
[proc_UpdateListItemWorkflowLock method](#) 120
[proc_UpdateListViewFormWebPartSource method](#) 121

[proc_UpdateListViewToDataViewForSite method](#) 121
[proc_UpdateListViewToDataViewForWeb method](#) 122
[proc_UpdateSiteResourceUsage method](#) 122
[proc_UpdateSolution method](#) 123
[proc_UpdateSolutionResourceUsage method](#) 124
[proc_UpdateViewWhileSaving method](#) 124
[proc_UpdateWebPart method](#) 126
[proc_UpdateWebPartCache method](#) 128
[proc_UpdateWebPartIsIncluded method](#) 129
[proc_UpdateWebPartProps method](#) 130
[proc_UpdateWebPartTypeId method](#) 132
[proc_UpdateWebPartWhileSaving method](#) 132
[proc_UpdateWorkflowAssociation method](#) 134
[proc_UpdateWorkItem method](#) 136
[proc_WorkflowHasVisibleParentItem method](#) 137
[sequencing rules](#) 31
[timer events](#) 137
[timers](#) 31
 Simple data types
[context collection identifier](#) 16
[context identifier](#) 16
[context object identifier](#) 16
[context type identifier](#) 16
[event receiver source identifier](#) 16
[list item version](#) 16
[workflow template identifier](#) 16
 Simple types - overview 26
[Solution Resource Usage Processing result set](#) 21
[Solution Resource Usage result set](#) 21
[Standards assignments](#) 15
 Structures
[binary](#) 19
[table and view](#) 26
[XML](#) 26

T

[Table structures - overview](#) 26

Timer events

[client](#) 138
[server](#) 137

Timers

[client](#) 138
[server](#) 31

[Tracking changes](#) 150

[Transport](#) 16

Types

[complex](#) 26
[simple](#) 26

U

[Update an event receiver example](#) 139

V

[Vendor-extensible fields](#) 13

[Versioning](#) 13

[View structures - overview](#) 26

W

[Web Part operations overview](#) 12
[Web Parts result set](#) 22
[Work item operations overview](#) 12
[Work Items result set](#) 25
[Workflow association configuration bit field](#) 17
[Workflow Associations result set](#) 23
[Workflow internal state bit field](#) 18
Workflow Modifications
 [element](#) 26
[Workflow operations overview](#) 12
[Workflow Status1 bit field](#) 18
[Workflow template identifier simple type](#) 16

X

[XML structures](#) 26