[MS-SSPSJ]:
SSP Scheduled Jobs Stored Procedures 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
04/04/2008	0.1		Initial Availability
06/27/2008	1.0	Major	Revised and edited the technical content
12/12/2008	1.01	Editorial	Revised and edited the technical content
07/13/2009	1.02	Major	Changes made for template compliance
08/28/2009	1.03	Editorial	Revised and edited the technical content
11/06/2009	1.04	Editorial	Revised and edited the technical content
02/19/2010	2.0	Editorial	Revised and edited the technical content
03/31/2010	2.01	Editorial	Revised and edited the technical content
04/30/2010	2.02	Editorial	Revised and edited the technical content
06/07/2010	2.03	Editorial	Revised and edited the technical content
06/29/2010	2.04	Minor	Clarified the meaning of the technical content.
07/23/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
09/27/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
06/10/2011	2.04	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction 5
1.1 Glossary 5
1.2 References 5
1.2.1 Normative References 5
1.2.2 Informative References 6
1.3 Protocol Overview (Synopsis) 6
1.4 Relationship to Other Protocols 7
1.5 Prerequisites/Preconditions 7
1.6 Applicability Statement 8
1.7 Versioning and Capability Negotiation 8
1.8 Vendor-Extensible Fields 8
1.9 Standards Assignments 8

2 Messages 9
2.1 Transport 9
2.2 Common Data Types 9
2.2.1 Simple Data Types and Enumerations 9
2.2.2 Assembly 9
2.2.3 Class 9
2.2.4 Job Data 11
2.2.5 Recurrence 11
2.2.6 Common Result Sets 12
2.2.6.1 Scheduled Job Result Set 12

3 Protocol Details 14
3.1 Protocol Server Details 14
3.1.1 Abstract Data Model 14
3.1.2 Timers 14
3.1.3 Initialization 14
3.1.4 Message Processing Events and Sequencing Rules 14
3.1.4.1 proc_MIP_AddScheduledJob 15
3.1.4.2 proc_MIP_GetScheduledJobs 16
3.1.4.2.1 Scheduled Job Result Set 16
3.1.4.3 proc_MIP_GetScheduledJobById 16
3.1.4.3.1 Scheduled Job Result Set 16
3.1.4.4 proc_MIP_GetScheduledJobsInInterval 17
3.1.4.4.1 Scheduled Job Result Set 17
3.1.4.5 proc_MIP_ModifyScheduledJob 17
3.1.4.6 proc_MIP_RefreshScheduledJob 18
3.1.4.7 proc_MIP_RemoveScheduledJob 19
3.1.5 Timer Events 19
3.1.6 Other Local Events 19
3.2 Protocol Client Details 19
3.2.1 Abstract Data Model 19
3.2.2 Timers 20
3.2.3 Initialization 20
3.2.4 Message Processing Events and Sequencing Rules 20
3.2.4.1 proc_MIP_GetScheduledJobsInInterval 20
3.2.5 Timer Events 21
3.2.6 Other Local Events 21

4 Protocol Examples 22
4.1 Schedule a User Profile Import Operation to Occur Daily at 1 AM 22
4.2 Change the Schedule for the User Profile Import Operation to Occur Daily at 6 AM 23
4.3 Delete the Scheduled User Profile Import Operation 25

5 Security 26
5.1 Security Considerations for Implementers 26
5.2 Index of Security Parameters 26

6 Appendix A: Product Behavior 27

7 Change Tracking 28

8 Index 29

1 Introduction

The Shared Services Provider Scheduled Jobs protocol allows clients to store information about logical units of functionality on the server, and query the server for actual execution at predefined intervals.

1.1 Glossary

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

Coordinated Universal Time (UTC)
GUID
Security Support Provider Interface (SSPI)

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

audience compilation
change token
colleague
distribution list
result set
return code
Shared Services Provider (SSP)
stored procedure
Structured Query Language (SQL)
Transact-Structured Query Language (T-SQL)
user profile
user profile change event

The following terms are specific to this document:

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

1.2 References

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

[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

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

[MS-UPSAUD] Microsoft Corporation, "User Profile Service Audiences Protocol Specification"

[MS-UPSGRAD] Microsoft Corporation, "User Profile Service Push Protocol Specification"

[MS-UPSIMP] Microsoft Corporation, "User Profile Import Protocol Specification"

[MS-UPSPROF] Microsoft Corporation, "User Profile Stored Procedures Protocol Specification"

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

1.2.2 Informative References

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

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

1.3 Protocol Overview (Synopsis)

This protocol allows clients to add, modify, refresh, and delete scheduled jobs from a store on the protocol server, as well as retrieve those scheduled jobs by using predefined criteria such as a unique identifier or jobs scheduled to be run within a specific interval. In addition, the protocol specifies the actions that the client takes when a particular scheduled job is retrieved and is to be executed in the next specified interval.

The following diagram shows the operations and data flow between the protocol client and the protocol server.

The **Add** operation adds a scheduled job to the database store on the server. The three **Get** operations use lookup criteria to retrieve scheduled jobs stored on the server. The **Modify** operation updates scheduled job information in the store. The **Refresh** operation updates the next occurrence for a scheduled job in the store. The **Remove** operation deletes a scheduled job from the store.

The client takes specific actions after invoking the **Get In Interval** operation, as depicted by the **Execute Scheduled Job** action in the diagram.

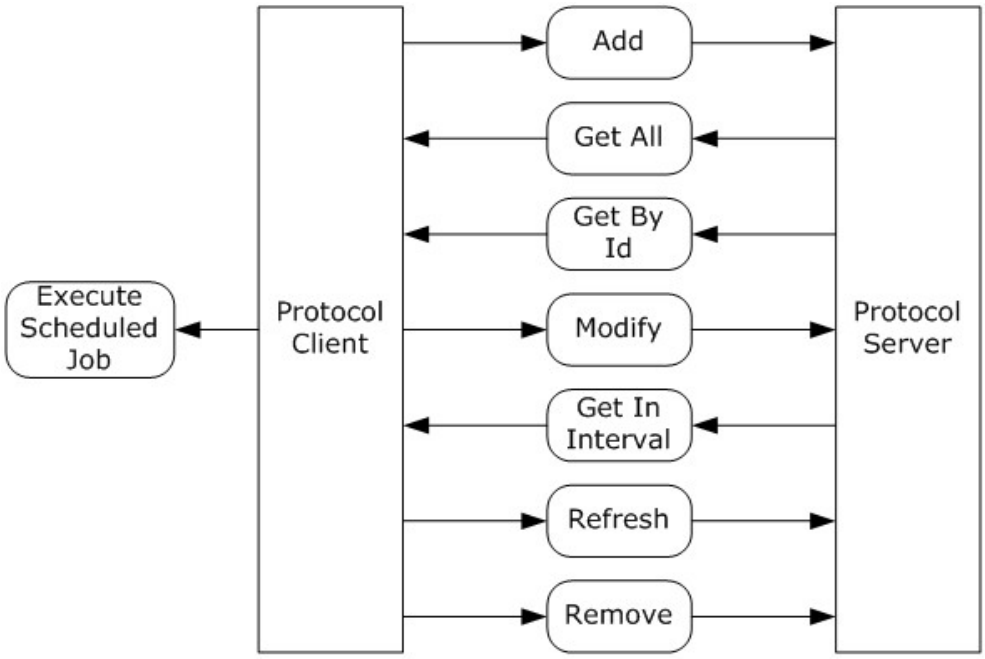

Figure 1: Operations and data flow diagram

1.4 Relationship to Other Protocols

The following diagram shows the transport stack for this protocol and the relationship to other protocols:

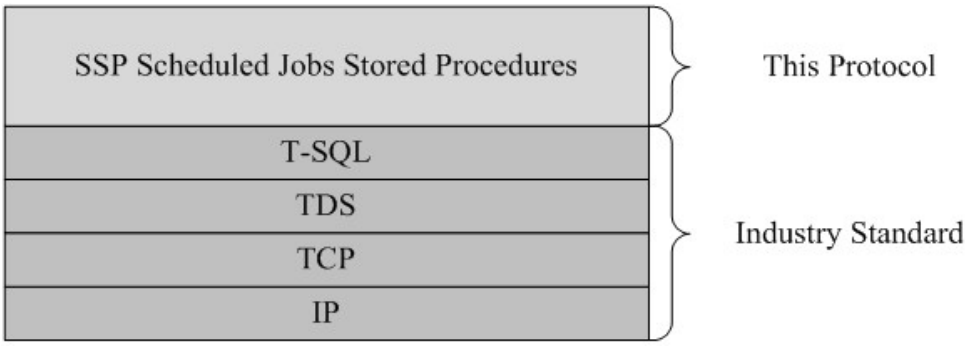

Figure 2: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

The operations described by this protocol operate between a protocol client and a protocol server. The client is expected to have the location and connection information for the required databases on the protocol server.

This protocol requires that the protocol client has appropriate permissions to call the **stored procedures** in the required databases on the protocol server.

1.6 Applicability Statement

This protocol is not designed to work with a generic set of scheduled jobs. This protocol can be used to work only with the jobs specified in this document.

1.7 Versioning and Capability Negotiation

Security and Authentication Methods: This protocol supports the **Security Support Provider Interface (SSPI)** and **SQL** authentication with the protocol server role specified in [MS-TDS].

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

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

2.2 Common Data Types

The following sections summarize the types that are defined in this protocol specification.

2.2.1 Simple Data Types and Enumerations

None.

2.2.2 Assembly

Assembly: nvarchar (256). Partly identifies the type of scheduled job. The value is in the following table.

Value	Description
Microsoft.Office.Project.Server.Administration, Version=12.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c	Ignored.
Microsoft.Office.Server, Version=12.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c	Partly identifies the actionable scheduled jobs.

If the value of **Assembly** is 'Microsoft.Office.Project.Server.Administration, Version=12.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c', then the value of **Class** is one of the following:

'Microsoft.Office.Project.Server.Administration.PWASSPSubmitSiteCreationTimerJob'

'Microsoft.Office.Project.Server.Administration.PWASSPSubmitServerScheduledTimerJob'

If the value of **Assembly** is 'Microsoft.Office.Server, Version=12.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c', then the value of **Class** is one of the following:

'Microsoft.Office.Server.UserProfiles.UserProfileChangeJob'

'Microsoft.Office.Server.UserProfiles.UserProfileChangeCleanupJob'

'Microsoft.Office.Server.UserProfiles.UserProfileImportJob'

'Microsoft.Office.Server.UserProfiles.UserProfileGradualUpgradeChangeJob'

'Microsoft.Office.Server.UserProfiles.DLImportJob'

'Microsoft.Office.Server.Audience.AudienceCompilationJob'

2.2.3 Class

Class: nvarchar (256). Partly identifies the type of a scheduled job. The value is in the following table.

Value	Description
Microsoft.Office.Project.Server.Administration.PWASSPSubmitSiteCreationTimerJob	Ignored.
Microsoft.Office.Project.Server.Administration.PWASSPSubmitServerScheduledTimerJob	Ignored.
Microsoft.Office.Server.UserProfiles.UserProfileChangeJob	Identifies the scheduled job that triggers an update of **colleagues** and generates anniversary events as specified in [MS-UPSPROF].
Microsoft.Office.Server.UserProfiles.UserProfileChangeCleanupJob	Identifies the scheduled job that triggers a deletion of **user profile change events** as specified in [MS-UPSPROF].
Microsoft.Office.Server.UserProfiles.UserProfileImportJob	Identifies the scheduled job which triggers the import of **user profiles** as specified in [MS-UPSIMP].
Microsoft.Office.Server.UserProfiles.UserProfileGradualUpgradeChangeJob	Identifies the scheduled job which triggers a sync of user profile changes as specified in [MS-UPSGRAD].
Microsoft.Office.Server.UserProfiles.DLImportJob	Identifies the scheduled job that triggers an incremental import of **distribution lists** as specified in [MS-UPSIMP].
Microsoft.Office.Server.Audience.AudienceCompilationJob	Identifies the scheduled job that triggers a full **audience compilation** as specified in [MS-UPSAUD].

If the value of **Class** is 'Microsoft.Office.Project.Server.Administration.PWASSPSubmitSiteCreationTimerJob' or 'Microsoft.Office.Project.Server.Administration.PWASSPSubmitServerScheduledTimerJob', then the value of **Assembly** is 'Microsoft.Office.Project.Server.Administration, Version=12.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c'.

If the value of **Class** is 'Microsoft.Office.Server.UserProfiles.UserProfileChangeJob',

'Microsoft.Office.Server.UserProfiles.UserProfileChangeCleanupJob', 'Microsoft.Office.Server.UserProfiles.UserProfileImportJob', 'Microsoft.Office.Server.UserProfiles.UserProfileGradualUpgradeChangeJob', 'Microsoft.Office.Server.UserProfiles.DLImportJob', or 'Microsoft.Office.Server.Audience.AudienceCompilationJob', then the value of **Assembly** is 'Microsoft.Office.Server, Version=12.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c'.

2.2.4 Job Data

Job Data (ntext): Information affecting the action taken by a scheduled job. The value MUST be either NULL or empty or one of the entries in the following table.

Value	Description
IsIncremental#True	Signals incremental import of user profiles to the scheduled job, as specified in [MS-UPSIMP].
IsIncremental#False	Signals full import of user profiles to the scheduled job, as specified in [MS-UPSIMP].
Flag;Date;ChangeToken	Indicates the type of profile change to the scheduled job, as specified in [MS-UPSGRAD]. Each element (*Flag*, *Date*, and *ChangeToken*) is replaced by an appropriate string, as follows: *Flag* indicates whether the job is currently running. The value MUST be 0 or 1, where 1 means the job is running. *Date* indicates the last run time of the job. The value MUST be a **datetime** value in **UTC**. *ChangeToken* indicates the type of change in the profile. The value MUST be a valid **change token**, as specified in [MS-UPSGRAD].

If the value of **Class** is 'Microsoft.Office.Server.UserProfiles.UserProfileImportJob', then the value of **Job Data** MUST be either 'IsIncremental#True' or 'IsIncremental#False'.

If the value of **Class** is not 'Microsoft.Office.Server.UserProfiles.UserProfileImportJob' or 'Microsoft.Office.Server.UserProfiles.UserProfileGradualUpgradeChangeJob', then the value of **Job Data** MUST be NULL or empty.

2.2.5 Recurrence

Recurrence (nvarchar(64)): The frequency with which a scheduled job is to be executed. The value MUST be in a format specified by the following set of rules in ABNF [RFC4234].

```
Recurrence = SecondRecurrence / MinuteRecurrence / HourlyRecurrence / DailyRecurrence/
WeeklyRecurrence / MonthlyRecurrence / YearlyRecurrence
SecondRecurrence = "every " ONETOFIFTYNINE " seconds"
MinuteRecurrence = "every " ONETOFIFTYNINE " minutes " SecondOrMinuteInstance
HourlyRecurrence = "hourly " SecondOrMinuteInstance
```

```
DailyRecurrence = "daily " TimeInstance
WeeklyRecurrence = "weekly " DayInstance
MonthlyRecurrence = "monthly " DateInstance
YearlyRecurrence = "yearly " MonthInstance
MonthInstance = ("at " MONTH " " DATE " " HourInstance) / ("between " MONTH " " DATE " "
HourInstance " and " MONTH " " DATE " " HourInstance)
DateInstance = ("at " DATE " " HourInstance) / ("between " DATE " " HourInstance " and " DATE
" " HourInstance)
DayInstance = ("at " DAY " " HourInstance) / ("between " DAY " " HourInstance " and " DAY " "
HourInstance)
TimeInstance = ("at " HourInstance) / ("between " HourInstance " and " HourInstance)
SecondOrMinuteInstance = ("at " ZEROTOFIFTYNINE) / ("between " ZEROTOFIFTYNINE " and "
ZEROTOFIFTYNINE)
HourInstance = HOUR ":" ZEROTOFIFTYNINE ":" ZEROTOFIFTYNINE
MONTH = "jan" / "feb" / "mar" / "apr" / "may" / "jun"/ "jul" / "aug" / "sep" / "oct" / "nov"
/ "dec"
DATE = %x30 %x31-39 / %x31-32 %x30-39 / %x33 %x30 / %x33 %x31 ; 01-31
DAY = "sun" / "mon" / "tue" / "wed" / "thu" / "fri" / "sat" / "su" / "mo" / "tu" / "we" /
"th" / "fr" / "sa"
HOUR = %x30-31 %x30-39 / %32 %x30-33 ; 00-23
ONETOFIFTYNINE = %x30 %x31-39 / %x31-35 %x30-39 ; 01-59
ZEROTOFIFTYNINE = %x30-35 %x30-39 ; 00-59
```

See Protocol Examples for examples with the *@Recurrence* parameter.

2.2.6 Common Result Sets

2.2.6.1 Scheduled Job Result Set

The scheduled job result set is defined by using **T-SQL**, as follows.

```
Assembly          nvarchar(256),
Class             nvarchar(256),
JobId             uniqueidentifier,
Recurrence        nvarchar(64),
JobDatan          text,
NextDueTime       datetime,
Disabled          bit,
DisplayName       nvarchar(256);
```

Assembly: A partial identifier of the type of scheduled job. The value MUST be an Assembly data type.

Class: A partial identifier of the type of scheduled job. The value MUST be a Class data type.

JobId: The identifier of the scheduled job.

Recurrence: The frequency of execution of the scheduled job. For nonrecurring (one-time) jobs, the value MUST be NULL. For recurring jobs, the value MUST be in the format specified in Recurrence.

JobData: The information affecting the action taken by a scheduled job. The value MUST be a Job Data data type.

NextDueTime: The next time the job is scheduled to be run. The value MUST be in UTC. If the returned value of **Disabled** is 1, the value of **NextDueTime** MUST be equivalent to 'Dec 31 9999 11:59:59:997 PM'.

Disabled: A flag indicating whether the scheduled job is disabled. The value MUST be in the following table.

Value	Description
0	The job is disabled.
1	The job is enabled.

DisplayName: A user friendly display name of the scheduled job.

3 Protocol Details

3.1 Protocol Server Details

3.1.1 Abstract Data Model

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

The server side of this protocol stores all implementation-specific information about scheduled jobs. The pieces of information maintained across message calls include a combination of identifiers to uniquely identify the job, the frequency of execution, and the next occurrence of the job. Scheduled jobs could be nonrecurring (one-time) jobs or recurring jobs. If the job is nonrecurring, the frequency of execution is NULL.

The server handles all message processing events for this protocol by manipulating the information for each scheduled job. An **Add** operation inserts a scheduled job along with its relevant information into the database store. A **Get** operation retrieves a scheduled job and its information based on the condition specified in the operation. Other operations such as **Modify**, **Refresh,** and **Remove** update or delete the information stored for the specified scheduled job.

3.1.2 Timers

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

3.1.3 Initialization

Before using this protocol, a connection that uses the underlying protocol layers specified in section 1.4, Relationship to Other Protocols, MUST be established as specified in [MS-TDS].

3.1.4 Message Processing Events and Sequencing Rules

This section describes the following stored procedures.

Procedure Name	Description
proc_MIP_AddScheduledJob	Adds a scheduled job to the store of existing jobs.
proc_MIP_GetScheduledJobs	Retrieves all scheduled jobs from the store of jobs.
proc_MIP_GetScheduledJobById	Retrieves a scheduled job by using a unique identifier from the store of jobs.
proc_MIP_GetScheduledJobsInInterval	Retrieves the set of scheduled jobs scheduled to be run before a specified time.
proc_MIP_ModifyScheduledJob	Modifies the information about a scheduled job in the store of existing jobs.
proc_MIP_RefreshScheduledJob	Updates the next occurrence of an existing scheduled job.

Procedure Name	Description
proc_MIP_RemoveScheduledJob	Removes a scheduled job from the store of existing jobs.

3.1.4.1 proc_MIP_AddScheduledJob

The **proc_MIP_AddScheduledJob** stored procedure is called to add a scheduled job to the store of scheduled jobs on the protocol server.

The T-SQL syntax for the stored procedure is as follows.

```
PROCEDURE proc_MIP_AddScheduledJob (
    @JobId            uniqueidentifier,
    @Assembly         nvarchar(256),
    @Class            nvarchar(256),
    @Recurrence       nvarchar(64) = NULL,
    @JobData          ntext = NULL,
    @NextDueTime      datetime,
    @Disabled         bit = 0,
    @DisplayName      nvarchar(256) = NULL
);
```

@JobId: The identifier of the scheduled job. The value MUST be a newly generated **GUID**.

@Assembly: A partial identifier of the type of scheduled job. The value MUST be an Assembly data type.

@Class: A partial identifier of the type of scheduled job. The value MUST be a Class data type.

@Recurrence: The frequency of execution of the scheduled job. For nonrecurring (one-time) jobs, the value MUST be NULL. For recurring jobs, the value MUST be in the format specified in Recurrence.

@JobData: The information affecting the action taken by a scheduled job. The value MUST be a Job Data data type.

@NextDueTime: The next time the job is scheduled to be run. The value MUST be in UTC. If the value of *@Disabled* is 1, the *@NextDueTime* value MUST be equivalent to 'Dec 31 9999 11:59:59:997 PM'.

@Disabled: A flag indicating whether the scheduled job is disabled. The value MUST be in the following table.

Value	Description
0	The job is enabled.
1	The job is disabled.

@DisplayName: A user friendly display name of the scheduled job.

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

Value	Description
0	Successful execution.
1	Failed execution.

Result Sets: MUST NOT return any result sets.

3.1.4.2 proc_MIP_GetScheduledJobs

The **proc_MIP_GetScheduledJobs** stored procedure is called to retrieve all scheduled jobs from the table of jobs.

The T-SQL syntax for the stored procedure is as follows.

```
PROCEDURE proc_MIP_GetScheduledJobs ();
```

Return Code Values: Returns an integer that MUST be 0.

Result Sets: MUST return one result set, described in the following section.

3.1.4.2.1 Scheduled Job Result Set

Returns an unordered list of all scheduled jobs stored on the protocol server. This result set MUST always be returned and MUST contain a row for each scheduled job stored on the protocol server. The result set is specified in Scheduled Job Result Set.

3.1.4.3 proc_MIP_GetScheduledJobById

The **proc_MIP_GetScheduledJobById** stored procedure is called to retrieve a scheduled job by using a unique identifier from the scheduled jobs stored on the protocol server.

The T-SQL syntax for the stored procedure is as follows.

```
PROCEDURE proc_MIP_GetScheduledJobById (
    @JobId              uniqueidentifier
);
```

@JobId: The identifier of the scheduled job to be retrieved.

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

Value	Description
0	Successful execution.
1	Failed execution.

Result Sets: MUST return one result set, described in the following section.

3.1.4.3.1 Scheduled Job Result Set

Returns the scheduled job with the specified *@JobId* from the store of scheduled jobs on the protocol server. The result set MUST always be returned. If the store on the protocol server contains

a scheduled job with the specified identifier, this result set MUST contain one row for that job; otherwise, the result set MUST be empty. The result set is specified in Scheduled Job Result Set.

3.1.4.4 proc_MIP_GetScheduledJobsInInterval

The **proc_MIP_GetScheduledJobsInInterval** stored procedure is called to retrieve the set of scheduled jobs from the store on the protocol server that are scheduled to be run before the specified *datetime* parameter.

Then, the protocol server MUST delete from the store all one-time occurring jobs that are returned, which are identified by their recurrence field being NULL.

The T-SQL syntax for the stored procedure is as follows.

```
PROCEDURE proc_MIP_GetScheduledJobsInInterval (
    @NextDueTime              datetime
);
```

@NextDueTime: The next time before which the scheduled jobs to be retrieved from the store are scheduled to be run.

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

Value	Description
0	Successful execution.
1	Failed execution.

Result Sets: MUST return one result set, described in the following section.

3.1.4.4.1 Scheduled Job Result Set

Returns the list of scheduled jobs that have the next scheduled occurrence before the specified time. This result set MUST be returned and MUST contain as many rows as there are scheduled jobs in the store matching the requirement. The result set is specified in Scheduled Job Result Set.

3.1.4.5 proc_MIP_ModifyScheduledJob

The **proc_MIP_ModifyScheduledJob** stored procedure is called to modify a scheduled job in the store on the protocol server.

The T-SQL syntax for the stored procedure is as follows.

```
PROCEDURE proc_MIP_ModifyScheduledJob (
    @JobId                    uniqueidentifier,
    @Assembly                 nvarchar(256),
    @Class                    nvarchar(256),
    @Recurrence               nvarchar(64) = NULL,
    @JobData                  ntext = NULL,
    @NextDueTime              datetime,
    @Disabled                 bit = 0,
    @DisplayName              nvarchar(256) = NULL
);
```

@JobId: The identifier of the scheduled job.

@Assembly: A partial identifier of the type of scheduled job. The value MUST be an Assembly data type.

@Class: A partial identifier of the type of scheduled job. The value MUST be a Class data type.

@Recurrence: The frequency of execution of the scheduled job. For nonrecurring (one-time) jobs, the value MUST be NULL. For recurring jobs, the value MUST be in the format specified in Recurrence.

@JobData: The information affecting the action taken by a scheduled job. The value MUST be a Job Data data type.

@NextDueTime: The next time the job is scheduled to be run. The value MUST be in UTC. If the value of *@Disabled* is 1, the **NextDueTime** value MUST be equivalent to 'Dec 31 9999 11:59:59:997 PM'.

@Disabled: A flag indicating whether the scheduled job is disabled. The value MUST be in the following table.

Value	Description
0	The job is enabled.
1	The job is disabled.

@DisplayName: A user friendly display name of the scheduled job.

If the scheduled job identified by the *@JobId* is found in the store on the protocol server, the protocol server MUST update all information for this job as specified by the parameters to the stored procedure.

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

Value	Description
0	Successful execution.
1	Failed execution.

Result Sets: MUST NOT return any result sets.

3.1.4.6 proc_MIP_RefreshScheduledJob

The **proc_MIP_RefreshScheduledJob** stored procedure is called to update the next occurrence of an existing scheduled job.

The T-SQL syntax for the stored procedure is as follows.

```
PROCEDURE proc_MIP_RefreshScheduledJob (
    @JobId                  uniqueidentifier,
    @NextDueTime            datetime
);
```

@JobId: The identifier of the scheduled job.

@NextDueTime: The next time the job is scheduled to be run. The value MUST be in UTC and MUST be greater than the current UTC time.

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

Value	Description
0	Successful execution.
1	Failed execution.

Result Sets: MUST NOT return any result sets.

3.1.4.7 proc_MIP_RemoveScheduledJob

The **proc_MIP_RemoveScheduledJob** stored procedure is called to remove a scheduled job from the table of jobs.

The T-SQL syntax for the stored procedure is as follows.

```
PROCEDURE proc_MIP_RemoveScheduledJob (
    @JobId                    uniqueidentifier
);
```

@JobId: The identifier of the scheduled job to be removed from the store on the protocol server.

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

Value	Description
0	Successful execution.
1	Failed execution.

Result Sets: MUST NOT return any result sets.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

3.2 Protocol Client Details

3.2.1 Abstract Data Model

None.

3.2.2 Timers

The client creates a pending jobs timer of 60 seconds to query for pending scheduled jobs on the protocol server that are scheduled to be run within the next timer interval.

3.2.3 Initialization

None.

3.2.4 Message Processing Events and Sequencing Rules

The protocol client handles each stored procedure with the same basic processing method of calling the stored procedure and waiting for the return code and any result sets that are returned.

This section describes the additional client behavior when invoking some of the stored procedures listed in Message Processing Events and Sequencing Rules on the protocol server.

3.2.4.1 proc_MIP_GetScheduledJobsInInterval

For each row returned in the scheduled job result set specified in Scheduled Job Result Set, after invoking the **proc_MIP_GetScheduledJobsInInterval** stored procedure on the protocol server:

If **Disabled** is 0, and **Assembly** and **Class** are in the following table, then the row MUST be ignored by the client.

Assembly	Class
Microsoft.Office.Project.Server.Administration, Version=12.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c	Microsoft.Office.Project.Server.Administration.PWASSPSubmitSiteCreationTimerJob
Microsoft.Office.Project.Server.Administration, Version=12.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c	Microsoft.Office.Project.Server.Administration.PWASSPSubmitServerScheduledTimerJob

If **Disabled** is 0, and **Assembly** is 'Microsoft.Office.Server, Version=12.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c', and **Class** is 'Microsoft.Office.Server.UserProfiles.UserProfileChangeJob', then the client MUST update colleagues and generate anniversary events as specified in [MS-UPSPROF].

If **Disabled** is 0, and **Assembly** is 'Microsoft.Office.Server, Version=12.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c', and **Class** is 'Microsoft.Office.Server.UserProfiles.UserProfileChangeCleanupJob', then the client MUST delete user profile change event as specified in [MS-UPSPROF].

If **Disabled** is 0, and **Assembly** is 'Microsoft.Office.Server, Version=12.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c', and **Class** is 'Microsoft.Office.Server.UserProfiles.UserProfileGradualUpgradeChangeJob', then the client MUST trigger a sync of user profile changes as specified in [MS-UPSGRAD].

If **Disabled** is 0, and **Assembly** is 'Microsoft.Office.Server, Version=12.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c', **Class** is 'Microsoft.Office.Server.UserProfiles.UserProfileImportJob' and **JobData** is 'IsIncremental#False', then the client MUST begin a full import of user profiles as specified in [MS-UPSIMP].

If **Disabled** is 0, and **Assembly** is 'Microsoft.Office.Server, Version=12.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c', **Class** is 'Microsoft.Office.Server.UserProfiles.UserProfileImportJob' and **Job Data** is 'IsIncremental#True', then the client MUST begin an incremental import of user profiles as specified in [MS-UPSIMP].

If **Disabled** is 0, and **Assembly** is 'Microsoft.Office.Server, Version=12.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c', and **Class** is 'Microsoft.Office.Server.UserProfiles.DLImportJob', then the client MUST begin an incremental import of distribution lists as specified in [MS-UPSIMP].

If **Disabled** is 0, and **Assembly** is 'Microsoft.Office.Server, Version=12.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c', and **Class** is 'Microsoft.Office.Server.Audience.AudienceCompilationJob', then the client MUST begin a full audience compilation as specified in [MS-UPSAUD].

The client MUST call the **proc_MIP_RefreshScheduledJob** stored procedure to update the next occurrence for the scheduled job on the protocol server. If **Disabled** is 1, the *@NextDueTime* value passed in to the stored procedure MUST be equivalent to 'Dec 31 9999 11:59:59:997 PM'.

3.2.5 Timer Events

The pending jobs timer specified in Timers triggers the following sequence of events:

The protocol client calls the proc_MIP_GetScheduledJobsInInterval stored procedure on the protocol server, as specified in section 3.1.4.4.

The protocol client processes the result set as specified in section 3.2.4.1.

3.2.6 Other Local Events

None.

4 Protocol Examples

This section provides specific example scenarios for sample scheduled job operations. These examples describe in detail the process of communication between the protocol server and protocol client. In conjunction with the detailed protocol documentation described in the reference documents, this information is intended to provide a comprehensive view on how the protocol client operates with the protocol server when scheduling and executing such an operation.

4.1 Schedule a User Profile Import Operation to Occur Daily at 1 AM

This example describes the requests made when an **SSP** administrator creates a scheduled job to trigger a user profile import operation every day at 1 AM UTC.

This example assumes that (a) a timer is set up on the client that polls the protocol server every 60 seconds for new jobs to execute in the next time interval, and that (b) the store of scheduled jobs is empty.

In this example, steps 1 through 6 need to occur in the specified order. The following actions happen:

1. -- proc_MIP_AddScheduledJob -->
2. <-- return code ignored --
3. -- proc_MIP_GetScheduledJobsInInterval -->
4. <-- Process Scheduled Job Result Set --
5. -- proc_MIP_RefreshScheduledJob -->
6. <-- return code ignored –
7. The client creates a scheduled job and adds it to the store on the protocol server. It does this by calling the proc_MIP_AddScheduledJob stored procedure by using [MS-TDS]. Consider the following T-SQL syntax, which displays the parameters used to call this stored procedure:

```
exec dbo.proc_MIP_AddScheduledJob
@JobId = 'E252E760-7AFE-4AA4-9045-DA86CDF0DEF7',
@Assembly = Microsoft.Office.Server, Version=12.0.0.0, Culture=neutral,
     PublicKeyToken=71e9bce111e9429c ',
@Class = 'Microsoft.Office.Server.UserProfiles.UserProfileImportJob',
@Recurrence = 'daily between 01:00:00 and 01:00:00',
@NextDueTime = 'Jan 31 2008 01:00:00:000AM',
@JobData = N'IsIncremental#True',
@Disabled = 0,
@DisplayName = N'User Profile Incremental Import Job'
```

8. The protocol server returns a 0 return code for successful execution, which is ignored by the client.
9. The client timer fires and queries the protocol server for new scheduled jobs to execute in this interval. It does this by calling the proc_MIP_GetScheduledJobsInInterval stored procedure by using [MS-TDS]. Consider the following T-SQL, which displays the parameters used to call this stored procedure just before 1 AM UTC:

```
exec dbo.proc_MIP_GetScheduledJobsInInterval
@NextDueTime = 'Jan 31 2008 01:01:01:000AM'
```

10. The protocol server returns the scheduled job result set, which contains at least one row for the user profile import job. When the client processes the row returned by the protocol server, it triggers the user profile import operation. The scheduled job result set includes the eight fields in the following two tables. For descriptions of fields in the scheduled job result set, see section 2.2.6.1.

Assembly	Class	JobId	Recurrence
Microsoft.Office.Server, Version=12.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c	Microsoft.Office.Server.UserProfiles.UserProfileImportJob	E252E760-7AFE-4AA4-9045-DA86CDF0DEF7	daily between 01:00:00 and 01:00:00

JobData	NextDueTime	Disabled	DisplayName
IsIncremental#True	'Jan 31 2008 01:00:00:000AM'	0	User Profile Incremental Import Job

1. The client updates the next occurrence for all scheduled jobs returned in the scheduled job result set in step 4. It does that by calling the proc_MIP_RefreshScheduledJob stored procedure for each row in that result set. Consider the following T-SQL syntax, which displays the parameters used to call this stored procedure**:**

```
exec dbo.proc_MIP_RefreshScheduledJob
JobId = 'E252E760-7AFE-4AA4-9045-DA86CDF0DEF7',
@NextDueTime = 'Feb 01 2008 01:00:00:000AM'
```

2. The protocol server returns a 0 return code for successful execution, which is ignored by the client.

4.2 Change the Schedule for the User Profile Import Operation to Occur Daily at 6 AM

This example describes the requests made when an SSP administrator updates the schedule of an existing scheduled job to trigger a user profile import operation every day at 6 AM UTC.

This example assumes that (a) a user profile import operation is previously scheduled, and that (b) the client has the *@JobId* for the scheduled job.

In this example, steps 1 through 4 need to occur in the specified order. The following actions happen.

1. -- **proc_MIP_GetScheduledJobById** -->
2. <-- Process Scheduled Job Result Set –
3. -- **proc_MIP_ModifyScheduledJob** -->

4. <-- return code ignored –

5. The client queries the protocol server for information about the existing user profile import scheduled job. It does this by calling the proc_MIP_GetScheduledJobById stored procedure by using [MS-TDS]. Consider the following T-SQL syntax, which displays the parameters used to call this stored procedure.

```
exec dbo.proc_MIP_GetScheduledJobById
@JobId = 'E252E760-7AFE-4AA4-9045-DA86CDF0DEF7'
```

6. The protocol server returns the scheduled job result set, which contains one row for the existing user profile import job, if it existed in the store on the protocol server; otherwise, the result set is empty. The client processes the row by retrieving the relevant information for the job and sets parameters for the next stored procedure call. The scheduled job result set includes the eight fields in the following two tables. For descriptions of fields in the scheduled job result set, see section 2.2.6.1.

Assembly	Class	JobId	Recurrence
Microsoft.Office.Server, Version=12.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c	Microsoft.Office.Server.UserProfiles.UserProfileImportJob	E252E760-7AFE-4AA4-9045-DA86CDF0DEF7	daily between 01:00:00 and 01:00:00

JobData	NextDueTime	Disabled	DisplayName
IsIncremental#True	'Jan 31 2008 01:00:00:000AM'	0	User Profile Incremental Import Job

1. The client updates the schedule for the user profile import scheduled job on the protocol server. It does this by calling the proc_MIP_ModifyScheduledJob stored procedure by using [MS-TDS]. Consider the following T-SQL syntax, which displays the parameters used to call this stored procedure.

```
exec dbo.proc_MIP_ModifyScheduledJob
@JobId = 'E252E760-7AFE-4AA4-9045-DA86CDF0DEF7',
@Assembly = Microsoft.Office.Server, Version=12.0.0.0, Culture=neutral,
   PublicKeyToken=71e9bce111e9429c ',
@Class = 'Microsoft.Office.Server.UserProfiles.UserProfileImportJob',
@Recurrence = 'daily between 06:00:00 and 06:00:00',
@NextDueTime = Feb 01 2008 06:00:00:000AM',
@JobData = N'IsIncremental#True',
@Disabled = 0,
@DisplayName = N'User Profile Incremental Import Job'
```

2. The protocol server returns a 0 return code for successful execution, which is ignored by the client.

4.3 Delete the Scheduled User Profile Import Operation

This example describes the requests made when an SSP administrator deletes a previously created scheduled job.

For simplicity, this example assumes that (a) a user profile import operation is previously scheduled by the admin, and that (b) the client has the *@JobId* for this scheduled job.

In this example, steps 1 through 2 need to occur in the specified order. The following actions happen.

1. -- proc_MIP_RemoveScheduledJob -->

   The client deletes the user profile import scheduled job from the store on the protocol server. It does this by calling the proc_MIP_RemoveScheduledJob stored procedure by using [MS-TDS]. Consider the following T-SQL, which displays the parameters used to call this stored procedure:

```
exec dbo.proc_MIP_RemoveScheduledJob
@JobId = 'E252E760-7AFE-4AA4-9045-DA86CDF0DEF7'
```

2. <-- return code ignored –

   The server returns a 0 return code for successful execution, which is ignored by the client.

5 Security

5.1 Security Considerations for Implementers

Interactions with SQL are susceptible to tampering and other forms of security risks. Implementers are advised to sanitize input parameters for stored procedures prior to invoking the stored procedure

There are no additional security considerations for implementers. Security assumptions of this protocol are documented in Prerequisites/Preconditions.

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® Office Forms Server 2007
- Microsoft® Office SharePoint® Server 2007
- Microsoft® SQL Server® 2005
- Microsoft® SQL Server® 2008
- Microsoft® SQL Server® 2008 R2

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

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

7 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

8 Index

A

Abstract data model
  client 19
  server 14
Applicability 8
Assembly common type 9

C

Capability negotiation 8
Change the Schedule for the User Profile Import Operation to Occur Daily at 6 AM example 23
Change tracking 28
Class common type 9
Client
  abstract data model 19
  initialization 20
  local events 21
  message processing 20
  proc_MIP_GetScheduledJobsInInterval method 20
  sequencing rules 20
  timer events 21
  timers 20
Common data types
  Assembly 9
  Class 9
  overview 9

D

Data model - abstract
  client 19
  server 14
Data types
  Assembly common type 9
  common 9
  Job Data 11
  Recurrence simple type 11
Data types - common
  Assembly 9
  Class (section 2.2.3 9, section 2.2.3 9)
Data types - simple
  Job Data 11
  overview 9
  Recurrence 11
Delete the Scheduled User Profile Import Operation example 25

E

Events
  local - client 21
  local - server 19
  timer - client 21
  timer - server 19
Examples
  Change the Schedule for the User Profile Import Operation to Occur Daily at 6 AM 23
  Delete the Scheduled User Profile Import Operation 25
  Schedule a User Profile Import Operation to Occur Daily at 1 AM 22

F

Fields - vendor-extensible 8

G

Glossary 5

I

Implementer - security considerations 26
Index of security parameters 26
Informative references 6
Initialization
  client 20
  server 14
Introduction 5

J

Job Data simple type 11

L

Local events
  client 21
  server 19

M

Message processing
  client 20
  server 14
Messages
  common data types 9
  enumerations 9
  Scheduled Job result set 12
  simple data types 9
  transport 9
Methods
  proc_MIP_AddScheduledJob 15
  proc_MIP_GetScheduledJobById 16
  proc_MIP_GetScheduledJobs 16
  proc_MIP_GetScheduledJobsInInterval (section 3.1.4.4 17, section 3.2.4.1 20)
  proc_MIP_ModifyScheduledJob 17
  proc_MIP_RefreshScheduledJob 18
  proc_MIP_RemoveScheduledJob 19

N

Normative references 5

O

Overview (synopsis) 6

P

Parameters - security index 26
Preconditions 7
Prerequisites 7
proc_MIP_AddScheduledJob method 15
proc_MIP_GetScheduledJobById method 16
proc_MIP_GetScheduledJobs method 16
proc_MIP_GetScheduledJobsInInterval method (section 3.1.4.4 17, section 3.2.4.1 20)
proc_MIP_ModifyScheduledJob method 17
proc_MIP_RefreshScheduledJob method 18
proc_MIP_RemoveScheduledJob method 19
Product behavior 27

R

Recurrence simple type 11
References
  informative 6
  normative 5
Relationship to other protocols 7
Result sets - messages
  Scheduled Job 12

S

Schedule a User Profile Import Operation to Occur
  Daily at 1 AM example 22
Scheduled Job result set 12
Security
  implementer considerations 26
  parameter index 26
Sequencing rules
  client 20
  server 14
Server
  abstract data model 14
  initialization 14
  local events 19
  message processing 14
  proc_MIP_AddScheduledJob method 15
  proc_MIP_GetScheduledJobById method 16
  proc_MIP_GetScheduledJobs method 16
  proc_MIP_GetScheduledJobsInInterval method 17
  proc_MIP_ModifyScheduledJob method 17
  proc_MIP_RefreshScheduledJob method 18
  proc_MIP_RemoveScheduledJob method 19
  sequencing rules 14
  timer events 19
  timers 14
Simple data types
  Job Data 11
  overview 9
  Recurrence 11
Standards assignments 8

T

Timer events
  client 21
  server 19
Timers
  client 20
  server 14
Tracking changes 28
Transport 9

V

Vendor-extensible fields 8
Versioning 8