[MS-SPSETSP]: SharePoint Subscription Settings Stored Procedures Protocol Specification

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	Minor	Updated 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	1.05	Minor	Clarified the meaning of the technical content.
09/27/2010	1.05	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	1.05	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	1.05	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	1.05	No change	No changes to the meaning, language, or formatting of the technical content.
06/10/2011	1.05	No change	No changes to the meaning, language, or formatting of the technical content.
01/20/2012	1.05	No change	No changes to the meaning, language, or formatting of the technical content.
04/11/2012	1.05	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	1.05	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction

The SharePoint Subscription Settings Stored Procedures Protocol allows for a set of configuration data to be shared among a set of site collections. This protocol is typically needed for hosting scenarios.

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

1.1 Glossary

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

GUID
property set

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

request identifier
root element
site collection
site subscription
Structured Query Language (SQL)
version stamp
XML fragment
XML namespace
XML namespace prefix
XML schema

The following terms are specific to this document:

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

1.2 References

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

1.2.1 Normative References

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

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

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

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

[XML10] World Wide Web Consortium, "Extensible Markup Language (XML) 1.0 (Third Edition)", February 2004, http://www.w3.org/TR/REC-xml

[XMLINFOSET] World Wide Web Consortium, "XML Information Set (Second Edition)", February 2004, http://www.w3.org/TR/2004/REC-xml-infoset-20040204

[XMLNS] Bray, T., Hollander, D., Layman, A., et al., Eds., "Namespaces in XML 1.0 (Third Edition)", W3C Recommendation, December 2009, http://www.w3.org/TR/2009/REC-xml-names-20091208/

[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".

1.3 Protocol Overview (Synopsis)

In enterprise and hosting scenarios a number of **site collections** need to use a shared set of configuration data. This protocol allows for the potential of some configuration data for this group of site collections to be delegated to additional administrators.

To facilitate these scenarios this protocol has the ability to set, remove and read **property sets** for a particular set of site collections.

1.4 Relationship to Other Protocols

This protocol uses T-SQL as shown in the following layering diagram:

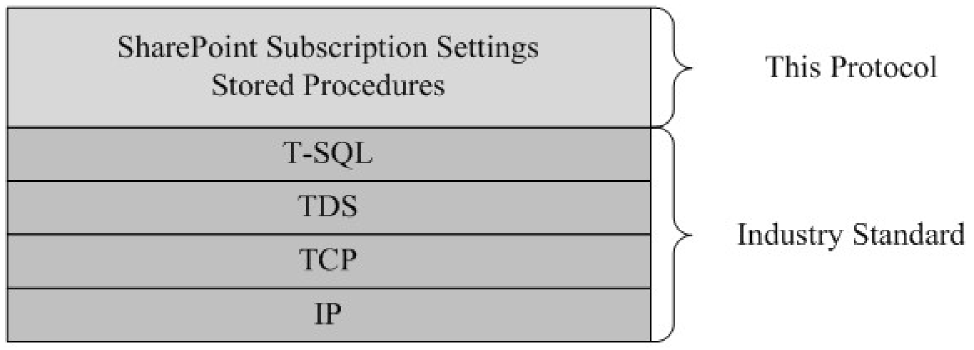

Figure 1: This protocol in relation to other protocols.

1.5 Prerequisites/Preconditions

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

This protocol requires that the protocol client has appropriate permissions to call the stored procedures in the back-end databases.

1.6 Applicability Statement

None.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

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

2.2 Common Data Types

The following sections define the common data types that are used in this protocol.

2.2.1 Subscription Settings Property Set XML

A Unicode string which MUST conform to the following **XML schema (2)**.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema id="object" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="entries">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="entry" minOccurs="0" maxOccurs="unbounded" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

entries: The **root element** of the **XML fragment**.

entry: A name/value pair. The semantic meaning of the pair is defined by sub-protocols which implement this protocol.

Each **entry** element MUST conform to one of the following XML schema (2)s:

2.2.1.1 nullEntry

An **entry** with no value.

```
<xs:complexType name="nullEntry">
  <xs:attribute name="name" type="xs:string" use="required" />
  <xs:attribute name="nil" type="xs:string" use="required" fixed="true" />
</xs:complexType>
```

name: The name of the **entry**.

nil: The attribute that indicates an entry with no value.

2.2.1.2 stringEntry

An **entry** that specifies a string value.

```
<xs:complexType name="stringEntry">
  <xs:simpleContent>
    <xs:extension base="xs:string" >
      <xs:attribute name="name" type="xs:string" use="required" />
```

```
        <xs:attribute name="type" type="xs:string" use="required" fixed="string" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
```

name: The name of the **entry**.

type: The type of the **entry**.

If the type identifier of a property set is "47EF919C-588D-4cfc-A552-762F746A5127", the value of this field MUST be a semi-colon delimited list of **GUID**s or an empty string.

2.2.1.3 intEntry

An **entry** that specifies an integer value.

```
  <xs:complexType name="intEntry">
    <xs:simpleContent>
      <xs:extension base="xs:int" >
        <xs:attribute name="name" type="xs:string" use="required" />
        <xs:attribute name="type" type="xs:string" use="required" fixed="int" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
```

name: The name of the **entry**.

type: The type of the **entry**.

2.2.1.4 longEntry

An **entry** that specifies a long value.

```
  <xs:complexType name="longEntry">
    <xs:simpleContent>
      <xs:extension base="xs:long" >
        <xs:attribute name="name" type="xs:string" use="required" />
        <xs:attribute name="type" type="xs:string" use="required" fixed="long" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
```

name: The name of the **entry**.

type: The type of the **entry**.

2.2.1.5 booleanEntry

An **entry** that specifies a Boolean value.

```
  <xs:complexType name="booleanEntry">
    <xs:simpleContent>
      <xs:extension base="xs:boolean" >
        <xs:attribute name="name" type="xs:string" use="required" />
        <xs:attribute name="type" type="xs:string" use="required" fixed="boolean" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
```

name: The name of the **entry**.

type: The type of the **entry**.

2.2.1.6 guidEntry

An **entry** that specifies a GUID value.

```
  <xs:simpleType name="guid">
    <xs:restriction base="xs:string">
      <xs:pattern value="\{[0-9A-F]{8}-[0-9A-F]{4}-[0-9A-F]{4}-[0-9A-F]{4}-[0-9A-F]{12}\}"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="guidEntry">
    <xs:simpleContent>
      <xs:extension base="guid" >
        <xs:attribute name="name" type="xs:string" use="required" />
        <xs:attribute name="type" type="xs:string" use="required" fixed="guid" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
```

name: The name of the **entry**.

type: The type of the **entry**.

2.2.1.7 datetimeEntry

An **entry** that specifies a **DateTime** value.

```
  <xs:complexType name="longEntry">
    <xs:simpleContent>
      <xs:extension base="xs:long" >
        <xs:attribute name="name" type="xs:string" use="required" />
        <xs:attribute name="type" type="xs:string" use="required" fixed="sp-dateTime" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
```

name: The name of the **entry**.

type: The type of the **entry**.

The value of this field MUST conform to the **DateTime** bit field as specified in section 2.2.3.1.

2.2.2 Simple Data Types and Enumerations

The following section defines the enumeration used in this protocol.

2.2.2.1 Subscription Settings Type Identifiers

The type identifier of a property set MUST be one of the values

Value	Description
D5C46399-6D04-4489-82CD-AA36B8ACCEFB	Properties
83E6C4CB-8E96-4882-AF00-421F586517F2	Administrative Properties
47EF919C-588D-4cfc-A552-762F746A5127	Feature Set

2.2.3 Bit Fields and Flag Structures

The following bit field is used by this protocol.

2.2.3.1 DateTime Bit Field

The **DateTime** bit field is a 64-bit value that MUST conform to the following model.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
Kind		Ticks																													
...																															

Kind (2 bits): The type of the **DateTime**. This value MUST be listed in the following table.

Value	Description
0x1	Coordinated Universal Time (UTC)
0x2	Local Time

Ticks (62 bits): The number of 100 nanosecond intervals that have elapsed between 12:00:00 midnight on January 1, 0001 and the date and time specified in Coordinated Universal Time (UTC).

2.2.4 Binary Structures

None.

2.2.5 Result Sets

2.2.5.1 proc_GetPropertySet.ResultSet0

This result set represents a property set stored by the protocol server.

```
Version bigint NOT NULL,
Xml xml NOT NULL,
PropertySetId uniqueidentifier NOT NULL,
```

Version: The **version stamp** of the property set.

Xml: The data associated with the property set. This value MUST conform to the Subscription Settings Property Set XML schema as specified in section 2.2.1.

PropertySetId: The unique identifier of the property set.

2.2.5.2 proc_GetPropertySetIdsByType.ResultSet0

This result set represents a set of unique identifiers that correspond to property sets stored by the protocol server. All identifiers in the result set MUST be unique, and MUST correspond to a different property set. Each property set MUST have the same type identifier.

```
PropertySetId uniqueidentifier NOT NULL,
```

PropertySetId: The unique identifier of the property set.

2.2.5.3 proc_GetSiteSubscriptions.ResultSet0

This result set contains a set of unique identifiers that represent **site subscriptions**. Each site subscription that appears in this result set MUST be unique.

```
SubscriptionId uniqueidentifier NOT NULL,
```

SubscriptionId: The unique identifier of the site subscription.

2.2.5.4 proc_GetSubscriptionPropertySetId.ResultSet0

This result set contains the unique identifier of the property set contained in the protocol server that matches the site subscription and type identifier specified when calling proc_GetSubscriptionPropertySetId. If the protocol server contains a property set matching the input parameters, this result set MUST contain one result. Otherwise, this result set MUST contain zero results.

```
PropertySetId uniqueidentifier NOT NULL,
```

PropertySetId: The unique identifier of the property set.

2.2.5.5 proc_SetSubscriptionMapping.ResultSet0

This result set reports information about the existence of a mapping between the specified site subscription and a property set of the specified type prior to execution of **proc_SetSubscriptionMapping**. If a pre-existing mapping was found, this result set MUST contain one result. If a pre-existing mapping was not found, this result set MUST contain zero results.

```
PropertySetId uniqueidentifier NOT NULL,
```

PropertySetId: The unique identifier of the property set.

2.2.5.6 proc_SetSubscriptionPropertySet.ResultSet0

This result set contains the version stamp of a property set.

```
Version bigint NOT NULL,
```

Version: The version stamp of an existing property set contained in the protocol server.

2.2.5.7 proc_GetSubscriptionPropertySet.proc_GetSubscriptionPropertySet.Default.ResultSet0

This result set contains the data and metadata associated with a property set.

```
Version bigint NOT NULL,
Xml xml NOT NULL,
PropertySetId uniqueidentifier NOT NULL,
```

Version: The version stamp of the property set stored by the protocol server.

Xml: The data associated with the property set. This value MUST conform to the Subscription Settings Property Set XML schema as specified in section 2.2.1.

PropertySetId: The unique identifier of the property set.

2.2.6 Tables and Views

2.2.6.1 PropertySet

The **PropertySet** table stores a list of property sets along with supporting metadata for each property set.

```
TypeId uniqueidentifier NOT NULL,
PropertySetId uniqueidentifier NOT NULL,
Version bigint NOT NULL,
Xml xml NULL,
```

TypeId: The type identifier of the property set. This value MUST exist in the list of valid Subscription Settings Type Identifiers as defined in section 2.2.2.1.

PropertySetId: The unique identifier of the property set.

Version: The version stamp of the property set.

Xml: The data associated with the property set. This value MUST conform to the Subscription Settings Property Set XML schema as specified in section 2.2.1.

2.2.6.2 Mapping

The **Mapping** table stores a list of mappings. Each mapping MUST contain the identifier of a site subscription, and the unique identifier and type identifier of a property set stored in the **PropertySet** table.

```
SubscriptionId uniqueidentifier NOT NULL,
TypeId uniqueidentifier NOT NULL,
PropertySetId uniqueidentifier NOT NULL,
```

SubscriptionId: The unique identifier of the site subscription.

TypeId: The type identifier of the property set. This value MUST exist in the list of valid Subscription Settings Type Identifiers as defined in section 2.2.2.1.

PropertySetId: The unique identifier of the property set.

2.2.7 XML Structures

None.

2.2.7.1 Namespaces

This protocol specifies and references **XML namespaces** using the mechanisms specified in [XMLNS]. Although this document associates a **XML namespace prefix** for each XML namespace that is used, the choice of any particular XML namespace prefix is implementation-specific and not significant for interoperability.

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

2.2.7.2 Simple Types

None.

2.2.7.3 Complex Types

None.

2.2.7.4 Elements

None.

2.2.7.5 Attributes

None.

2.2.7.6 Groups

None.

2.2.7.7 Attribute Groups

None.

3 Protocol Details

3.1 Server Details

The protocol server responds only to stored procedure calls from the protocol client. It returns result sets and return codes and never initiates communication with other endpoints of this protocol.

3.1.1 Abstract Data Model

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

The protocol server maintains a list of property sets and their mappings to site subscription identifiers. The protocol server adds, changes, and deletes property set objects and their mappings from this list in response to messages from the protocol client.

There are three types of property sets: Properties, Administrative Properties, and Feature Set. The Properties type is used to store a collection of settings, the Administrative Properties stores values specifically for the administrator of the system, and a Feature Set stores a list of identifiers that can be used to filter end user functionality. Each site subscription identifier can potentially be associated with one of each type of property set.

Every property set has exactly one version stamp, type identifier, and property set identifier. A property set is only uniquely identified by a combination of its property set identifier and type identifier.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

None.

3.1.5 Message Processing Events and Sequencing Rules

3.1.5.1 proc_DropPropertySet

This **proc_DropPropertySet** stored procedure deletes a property set and all associated mappings to the property set from the protocol server. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_DropPropertySet (
@PropertySetId uniqueidentifier,
@TypeId uniqueidentifier,
@Version bigint,
@CorrelationId uniqueidentifier = null,
```

);

@PropertySetId: The unique identifier of the property set. This value MUST NOT be NULL or empty.

@TypeId: The type identifier of the property set. This value MUST exist in the list of valid Subscription Settings Type Identifiers as defined in section 2.2.2.1.

@Version: The version stamp of the property set. This value MUST NOT be zero and MUST be non-negative.

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

Return values:

Value	Description
1	This value MUST be returned for all error conditions other than those specified in this table. If this value is returned, the protocol server MUST NOT delete the property set or any associated mappings.
2	This value MUST be returned if the value of **@Version** does not match the version stamp of the property set stored by the protocol server. If this value is returned, the protocol server MUST NOT delete the property set or any associated mappings.
3	This value MUST be returned if the values of **@PropertySetId** and **@TypeId** do not reference an existing property set. If this value is returned, the protocol server MUST NOT delete the property set or any associated mappings.
0	This value MUST be returned when the property set and associated mappings are deleted by the protocol server.

Result Sets: MUST NOT return any result sets.

3.1.5.2 proc_DropSubscriptionMapping

The **proc_DropSubscriptionMapping** stored procedure deletes a mapping between a site subscription and a property set of a specific type. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_DropSubscriptionMapping (
@SubscriptionId uniqueidentifier,
@TypeId uniqueidentifier,
@CorrelationId uniqueidentifier = null,
```

);

@SubscriptionId: The unique identifier of the site subscription. This value MUST NOT be NULL or empty.

@TypeId: The type identifier of the property set. This value MUST exist in the list of valid Subscription Settings Type Identifiers as defined in section 2.2.2.1.

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

Return values:

Value	Description
1	This value MUST be returned for all error conditions encountered by the protocol server. If this value is returned, the protocol server MUST NOT delete any mappings.
0	This value MUST be returned when the mapping of the specified type and to the specified site subscription is deleted or did not exist.

Result Sets: MUST NOT return any result sets.

3.1.5.3 proc_GetPropertySet

The **proc_GetPropertySet** stored procedure retrieves the property set represented by the specified unique identifier and type identifier. If the protocol server contains a matching property set, the result set MUST contain one result. If the protocol server does not contain a matching property set, the result set MUST contain zero results. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetPropertySet (
@PropertySetId uniqueidentifier,
@TypeId uniqueidentifier,
@CorrelationId uniqueidentifier = null,
```

);

@PropertySetId: The unique identifier of the property set. This value MUST NOT be NULL or empty.

@TypeId: The type identifier of the property set. This value MUST exist in the list of valid Subscription Settings Type Identifiers as defined in section 2.2.2.1.

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

Return values:

Value	Description
1	This value MUST be returned for all error conditions.
0	This value MUST be returned when the operation completes successfully, regardless of whether or not the result set contains any results.

Result Sets:

This procedure MUST return proc_GetPropertySet.ResultSet0

3.1.5.4 proc_GetPropertySetIdsByType

The **proc_GetPropertySetIdsByType** stored procedure retrieves the set of unique identifiers that represent property sets of a specific type stored by the protocol server. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetPropertySetIdsByType (
@TypeId uniqueidentifier,
@CorrelationId uniqueidentifier = null,
```

);

@TypeId: The type identifier of the property set. This value MUST exist in the list of valid Subscription Settings Type Identifiers as defined in section 2.2.2.1.

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

Return values:

Value	Description
1	This value MUST be returned for all error conditions.
0	This value MUST be returned if the operation completes successfully.

Result Sets:

This procedure MUST return proc_GetPropertySetIdsByType.ResultSet0

3.1.5.5 proc_GetSiteSubscriptions

The **proc_GetSiteSubscriptions** stored procedure retrieves a list of site subscriptions from the protocol server. The result set MUST contain one result for each unique site subscription that has one or more mappings to property sets stored by the protocol server. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetSiteSubscriptions (
@CorrelationId uniqueidentifier = null,
```

);

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

Return values:

Value	Description
1	This value MUST be returned for all error conditions.
0	This value MUST be returned if the operation completes successfully.

Result Sets:

This procedure MUST return proc_GetSiteSubscriptions.ResultSet0

3.1.5.6 proc_GetSubscriptionPropertySet

The **proc_GetSubscriptionPropertySet** stored procedure retrieves a single property set from the protocol server. The property set MUST match the type specified. The protocol server MUST contain a mapping between the specified site subscription and the property set. If these constraints cannot be met, then the stored procedure MUST return an empty result set. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetSubscriptionPropertySet (
@SubscriptionId uniqueidentifier,
@TypeId uniqueidentifier,
@CorrelationId uniqueidentifier = null,
```

);

@SubscriptionId: The unique identifier of the site subscription. This value MUST NOT be NULL or empty.

@TypeId: The type identifier of the property set. This value MUST exist in the list of valid Subscription Settings Type Identifiers as defined in section 2.2.2.1.

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

Return values:

Value	Description
1	This value MUST be returned for all error conditions.
0	This value MUST be returned if the operation completes successfully, regardless of whether or not the result set contains any results.

Result Sets:

This procedure MUST return a proc_GetSubscriptionPropertySet.proc_GetSubscriptionPropertySet.Default.ResultSet0

3.1.5.7 proc_GetSubscriptionPropertySetId

The **proc_GetSubscriptionPropertySetId** stored procedure retrieves the unique identifier of a single property set from the protocol server. The property set MUST match the type specified. The protocol server MUST contain a mapping between the specified site subscription and the property set. If these constraints cannot be met, then the stored procedure MUST return an empty result set. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_GetSubscriptionPropertySetId (
@SubscriptionId uniqueidentifier,
@TypeId uniqueidentifier,
@CorrelationId uniqueidentifier = null,
```

);

@SubscriptionId: The unique identifier of the site subscription. This value MUST NOT be NULL or empty.

@TypeId: The type identifier of the property set. This value MUST exist in the list of valid Subscription Settings Type Identifiers as defined in section 2.2.2.1.

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

Return values:

Value	Description
1	This value MUST be returned for all error conditions.
0	This value MUST be returned if the operation completes successfully, regardless of whether or not the result set contains any results.

Result Sets:

This procedure MUST return proc_GetSubscriptionPropertySetId.ResultSet0

3.1.5.8 proc_PutPropertySet

The **proc_PutPropertySet** stored procedure creates a new property set or updates an existing property set on the protocol server. If the unique identifier specified in **@PropertySetId** and the type identifier specified in **@TypeId** reference a property set stored by the protocol server, then the protocol server MUST update the property set. If the unique identifier specified in **@PropertySetId** and the type identifier specified in **@TypeId** do not reference a property set stored by the protocol server, then the protocol server MUST create a new property set. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_PutPropertySet (
@PropertySetId uniqueidentifier,
@TypeId uniqueidentifier,
@Version bigint,
@Xml xml,
@NewVersion bigint OUTPUT,
@CorrelationId uniqueidentifier = null,
```

);

@PropertySetId: The unique identifier of the property set. This value MUST NOT be NULL or empty.

@TypeId: The type identifier of the property set. This value MUST exist in the list of valid Subscription Settings Type Identifiers as defined in section 2.2.2.1.

@Version: The version stamp of the property set.

@Xml: The new data associated with the property set. The value MUST conform to the Subscription Settings Property Set XML schema as specified in section 2.2.1.

@NewVersion: The version stamp of the property set after it has been created or updated.

Possible parameter values:

Value	Description
1	This value MUST be returned as the value of **@NewVersion** if the stored procedure is creating a new property set on the protocol server.
PropertySet.Version	This value MUST be returned as the value of **@NewVersion** if the stored procedure is updating an existing property set on the protocol server. The value returned MUST be the value of **@Version** incremented by one. MUST be a value from column Version of table PropertySet

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

Return values:

Value	Description
1	This value MUST be returned when the value of **@Version** is not NULL, and **@PropertySetId** and **@TypeId** refer to a non-existent property set on the protocol server. If this value is returned, this stored procedure MUST NOT make any changes to the protocol server.
2	This value MUST be returned if the protocol server encounters an error storing a new property set. If this value is returned, this stored procedure MUST NOT make any changes to the protocol server.
3	This value MUST be returned when the protocol server attempts to update an existing property set, but the value of **@Version** is NULL or does not match the version stamp of the existing property set stored by the protocol server. If this value is returned, this stored procedure MUST NOT make any changes to the protocol server.
4	This value MUST be returned if the protocol server encounters an error updating an existing property set. If this value is returned, this stored procedure MUST NOT make any changes to the protocol server.
0	This value MUST be returned if the operation completes successfully.

Result Sets: MUST NOT return any result sets.

3.1.5.9 proc_SetSubscriptionMapping

The **proc_SetSubscriptionMapping** stored procedure creates or updates a mapping between a site subscription and a property set. If no mapping exists, then this stored procedure MUST create a new one. If a mapping does exist, then this stored procedure must overwrite it. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_SetSubscriptionMapping (
@SubscriptionId uniqueidentifier,
@TypeId uniqueidentifier,
@PropertySetId uniqueidentifier,
@CorrelationId uniqueidentifier = null,
```

);

@SubscriptionId: The unique identifier of the site subscription. This value MUST NOT be NULL or empty.

@TypeId: The type identifier of the property set. This value MUST exist in the list of valid Subscription Settings Type Identifiers as defined in section 2.2.2.1.

@PropertySetId: The unique identifier of the property set. This value MUST NOT be NULL or empty.

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

Return values:

Value	Description
1	This value MUST be returned if the protocol server encounters an error when attempting to create a new mapping between the specified site subscription and the specified property set. If this value is returned, this stored procedure MUST NOT make any changes to the protocol server.
2	This value MUST be returned if the protocol server encounters an error when attempting to update the existing mapping between the specified site subscription and a property set of the specified type. If this value is returned, this stored procedure MUST NOT make any changes to the protocol server.
0	This value MUST be returned if the operation completes successfully.

Result Sets:

This procedure MUST return proc_SetSubscriptionMapping.ResultSet0

3.1.5.10 proc_SetSubscriptionPropertySet

The **proc_SetSubscriptionPropertySet** stored procedure creates a new property set and a new mapping between the property set and a site subscription. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_SetSubscriptionPropertySet (
@SubscriptionId uniqueidentifier,
@TypeId uniqueidentifier,
@PropertySetId uniqueidentifier,
@Xml xml,
@NewVersion bigint OUTPUT,
@CorrelationId uniqueidentifier = null,
```

);

@SubscriptionId: The unique identifier of the site subscription. This value MUST NOT be NULL or empty.

@TypeId: The type identifier of the property set. This value MUST exist in the list of valid Subscription Settings Type Identifiers as defined in section 2.2.2.1.

@PropertySetId: The unique identifier of the property set. This value MUST NOT be NULL or empty.

@Xml: The new data associated with the property set. This value MUST conform to the Subscription Settings Property Set XML schema as specified in section 2.2.1.

@NewVersion: The version stamp of the property set after it has been created.

Possible parameter values:

Value	Description
1	This value MUST always be returned.

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

Return values:

Value	Description
1	This value MUST be returned if the protocol server encounters an error while checking for an existing property set with the unique identifier specified in **@PropertySetId** and type identifier specified in **@TypeId**. If this value is returned, this stored procedure MUST NOT make any changes to the protocol server.
2	This value MUST be returned if the protocol server contains an existing property set with the unique identifier specified in **@PropertySetId** and type identifier specified in **@TypeId**. If this value is returned, this stored procedure MUST NOT make any changes to the protocol server.
3	This value MUST be returned if the protocol server encounters an error while creating the property set. If this value is returned, this stored procedure MUST NOT make any changes to the protocol server.
4	This value MUST be returned if the protocol server encounters an error while checking for an existing mapping between the site subscription specified in **@SubscriptionId** and a property set of the type specified in **@TypeId**. If this value is returned, this stored procedure MUST NOT make any changes to the protocol server.
5	This value MUST be returned if the protocol server contains an existing mapping between the site subscription specified in **@SubscriptionId** and a property set of the type specified in **@TypeId**. If this value is returned, this stored procedure MUST NOT make any changes to the protocol server.
6	This value MUST be returned if the protocol server encounters an error while creating a mapping between the site subscription specified in **@SubscriptionId** and the new property set. If this value is returned, this stored procedure MUST NOT make any changes to the protocol server.
0	This value MUST be returned if the new property set is successfully created, and a new mapping is established between the site subscription specified in **@SubscriptionId** and the property set.

Result Sets:

This procedure MUST return proc_SetSubscriptionPropertySet.ResultSet0

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Client Details

None.

3.2.1 Abstract Data Model

None.

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Higher-Layer Triggered Events

None.

3.2.5 Message Processing Events and Sequencing Rules

None.

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

4 Protocol Examples

4.1 Manage the Feature Set of a Site Subscription

These examples illustrate the sequence of requests made and responses returned between the protocol client and protocol server during execution of feature set (a group of feature definition identifiers) management operations.

4.1.1 Create a Feature Set

To create a new feature set with a specific list of feature definition identifiers, the protocol client creates a property set, setting the type identifier to the value that denotes a feature set.

The protocol client creates a new property set of type feature set by calling the proc_PutPropertySet. It passes a unique identifier as the property set identifier (for example: "9C7C959D-DAF3-4054-BBF7-F2CFA81D20A5"), the type identifier ("47EF919C-588D-4CFC-A552-762F746A5127"), NULL as the version, an XML string that represents the feature definition identifiers (for example: `'<entries><entry name="FeatureIds" type="string"/></entries>'`), an out variable to collect the new version number, and a request identifier of the current request (for example: "00000000-0000-0000-0000-000000000000").

The protocol client calls **proc_PutPropertySet** as follows:

```
declare @p4 xml
set @p4=convert(xml,N'<entries><entry name="FeatureIds" type="string"/></entries>')
declare @p5 bigint
set @p5=NULL
exec dbo.proc_PutPropertySet @PropertySetId='9C7C959D-DAF3-4054-BBF7-F2CFA81D20A5',
@TypeId='47EF919C-588D-4CFC-A552-762F746A5127' , @Version=NULL, @Xml=@p4, @NewVersion=@p5
output, @CorrelationId='00000000-0000-0000-0000-000000000000'
```

4.1.2 Assign a Feature Set to a Site Subscription

After creating the feature set (previous example), the protocol client assigns the feature set to a site subscription.

The protocol client assigns an existing property set to a site subscription by calling proc_SetSubscriptionMapping. It passes the site subscription identifier (for example: "094EC36A-D30D-4E5E-8CD6-083187304A5A"), the type identifier ("47EF919C-588D-4CFC-A552-762F746A5127"), the property set identifier (for example: "9C7C959D-DAF3-4054-BBF7-F2CFA81D20A5"), and the request identifier (for example: "00000000-0000-0000-0000-000000000000").

The protocol client calls **proc_SetSubscriptionMapping** as follows:

```
exec dbo.proc_SetSubscriptionMapping @SubscriptionId='094EC36A-D30D-4E5E-8CD6-083187304A5A',
@TypeId='47EF919C-588D-4CFC-A552-762F746A5127', @PropertySetId='9C7C959D-DAF3-4054-BBF7-
F2CFA81D20A5', @CorrelationId='00000000-0000-0000-0000-000000000000'
```

4.1.3 Update the Feature Definition Identifiers of a Feature Set

To update the list of feature definition identifiers belonging to a feature set, the protocol client updates the current XML string by adding or removing feature definition identifiers, and then commits the change by calling proc_PutPropertySet with the updated feature definition identifier XML string.

The protocol client updates an existing property set of type feature set by calling **proc_PutPropertySet**. It passes the property set identifier (for example: "9C7C959D-DAF3-4054-BBF7-F2CFA81D20A5"), the type identifier ("47EF919C-588D-4CFC-A552-762F746A5127"), the current version, an XML string that represents the new set of feature definition identifiers (for example: `<entries><entry name="FeatureIds" type="string">3a1dd49f-c4a7-4294-a6ef-08ee1677010e;</entry></entries>'`), an out variable to collect the new version number, and the request identifier (for example: "00000000-0000-0000-0000-000000000000").

The protocol client calls **proc_PutPropertySet** as follows:

```
declare @p4 xml
set @p4=convert(xml,N'<entries><entry name="FeatureIds" type="string">3a1dd49f-c4a7-4294-
a6ef-08ee1677010e;</entry></entries>')
declare @p5 bigint
set @p5=NULL

exec dbo.proc_PutPropertySet @PropertySetId='9C7C959D-DAF3-4054-BBF7-F2CFA81D20A5',
@TypeId='47EF919C-588D-4CFC-A552-762F746A5127', @Version=1, @Xml=@p4, @NewVersion=@p5 output,
@CorrelationId='00000000-0000-0000-0000-000000000000'
```

4.1.4 Fetch All Existing Feature Sets

To retrieve all existing feature sets, the protocol client retrieves all property sets that have their type identifier set to that of a feature set.

The protocol client first calls proc_GetPropertySetIdsByType to get all the property set identifiers, belonging to property sets that have their type identifier set to that of a feature set. It passes the type identifier ("47EF919C-588D-4CFC-A552-762F746A5127") and the request identifier (for example: "00000000-0000-0000-0000-000000000000").

For each of the property set identifiers returned, the protocol client then calls **proc_GetPropertySet** to get the details of the property set. It passes the property set identifier (for example: "9C7C959D-DAF3-4054-BBF7-F2CFA81D20A5"), the type identifier ("47EF919C-588D-4CFC-A552-762F746A5127"), and the request identifier (for example: "00000000-0000-0000-0000-000000000000").

The protocol client calls **proc_GetPropertySetIdsByType** as follows:

```
exec dbo.proc_GetPropertySetIdsByType @TypeId='47EF919C-588D-4CFC-A552-
762F746A5127',@CorrelationId='00000000-0000-0000-0000-000000000000'
```

The result set returned is as shown in the following table.

PropertySetId
9C7C959D-DAF3-4054-BBF7-F2CFA81D20A5

The protocol client then calls **proc_GetPropertySet** for each of the property set identifiers returned to get details about the property set:

```
exec dbo.proc_GetPropertySet @PropertySetId='9C7C959D-DAF3-4054-BBF7-F2CFA81D20A5',
@TypeId='47EF919C-588D-4CFC-A552-762F746A5127',@CorrelationId='00000000-0000-0000-0000-
000000000000'
```

The result set returned is as shown in the following table.

Version	Xml	PropertySetId
2	<entries><entry name="FeatureIds" type="string" />3a1dd49f-c4a7-4294-a6ef-08ee1677010e;</entries>	9C7C959D-DAF3-4054-BBF7-F2CFA81D20A5

4.1.5 Fetch the Feature Set Assigned to a Site Subscription

The details of a property set that is assigned to a site subscription can be retrieved in one call to the protocol server.

The protocol client calls **proc_GetSubscriptionPropertySet** to get the property set assigned to a site subscription. It passes the site subscription identifier (for example: "094EC36A-D30D-4E5E-8CD6-083187304A5A"), the type identifier ("47EF919C-588D-4CFC-A552-762F746A5127"), and the request identifier (for example: "00000000-0000-0000-0000-000000000000").

The protocol client calls **proc_GetSubscriptionPropertySet** as follows:

```
exec dbo.proc_GetSubscriptionPropertySet @SubscriptionId='094EC36A-D30D-4E5E-8CD6-
083187304A5A',@TypeId='47EF919C-588D-4CFC-A552-762F746A5127',@CorrelationId='00000000-0000-
0000-0000-000000000000'
```

The result set returned is as shown in the following table.

Version	Xml	PropertySetId
2	<entries><entry name="FeatureIds" type="string" />3a1dd49f-c4a7-4294-a6ef-08ee1677010e;</entries>	9C7C959D-DAF3-4054-BBF7-F2CFA81D20A5

4.1.6 Delete a Feature Set

A property set can be deleted from the protocol server by calling **proc_DropPropertySet**.

The protocol client deletes a property set by calling **proc_DropPropertySet**. It passes the property set identifier (for example: "9C7C959D-DAF3-4054-BBF7-F2CFA81D20A5"), the type identifier ("47EF919C-588D-4CFC-A552-762F746A5127"), the current version, and the request identifier (for example: "00000000-0000-0000-0000-000000000000").

The protocol client calls **proc_DropPropertySet** as follows:

```
exec dbo.proc_DropPropertySet @PropertySetId='9C7C959D-DAF3-4054-BBF7-F2CFA81D20A5'
,@TypeId='47EF919C-588D-4CFC-A552-762F746A5127',@Version=2,@CorrelationId='00000000-0000-
0000-0000-000000000000'
```

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 before invoking them.

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

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