[MS-UPASP2]:
User Profile Admin Stored Procedures
Version 2 Protocol Specification

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.

- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL’s, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.

- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.

- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft’s delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft [Open Specification Promise](#) or the [Community Promise](#). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting [iplg@microsoft.com](mailto: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.
Preliminary Documentation. This Open Specification provides documentation for past and current releases and/or for the pre-release (beta) version of this technology. This Open Specification is final documentation for past or current releases as specifically noted in the document, as applicable; it is preliminary documentation for the pre-release (beta) versions. Microsoft will release final documentation in connection with the commercial release of the updated or new version of this technology. As the documentation may change between this preliminary version and the final version of this technology, there are risks in relying on preliminary documentation. To the extent that you incur additional development obligations or any other costs as a result of relying on this preliminary documentation, you do so at your own risk.

Revision Summary

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision History</th>
<th>Revision Class</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>01/20/2012</td>
<td>0.1</td>
<td>New</td>
<td>Released new document.</td>
</tr>
<tr>
<td>04/11/2012</td>
<td>0.1</td>
<td>No change</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>07/16/2012</td>
<td>0.1</td>
<td>No change</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
</tbody>
</table>
# 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 Overview ............................................................................................................. 6  
   1.4 Relationship to Other Protocols ......................................................................... 7  
   1.5 Prerequisites/Preconditions ................................................................................. 7  
   1.6 Applicability Statement ....................................................................................... 7  
   1.7 Versioning and Capability Negotiation ............................................................... 7  
   1.8 Vendor-Extensible Fields .................................................................................... 7  
   1.9 Standards Assignments ....................................................................................... 7  

2 Messages .................................................................................................................. 8  
   2.1 Transport ............................................................................................................. 8  
   2.2 Common Data Types ........................................................................................... 8  
      2.2.1 Simple Data Types and Enumerations ......................................................... 8  
      2.2.2 Bit Fields and Flag Structures ..................................................................... 8  
         2.2.2.1 User Rights Flags .................................................................................. 8  
      2.2.3 Binary Structures ....................................................................................... 8  
      2.2.4 Result Sets .................................................................................................. 8  
      2.2.4.1 Admin_ListPartitions.ResultSet0 ......................................................... 8  
      2.2.4.2 PartitionProperties ............................................................................... 9  
      2.2.5 Tables and Views ...................................................................................... 10  
      2.2.5.1 Tenants .................................................................................................. 10  
      2.2.6 XML Structures .......................................................................................... 11  
         2.2.6.1 Namespaces .......................................................................................... 11  
         2.2.6.2 Simple Types ...................................................................................... 11  
         2.2.6.3 Complex Types ................................................................................... 11  
            2.2.6.3.1 acl ............................................................................................... 12  
         2.2.6.4 Elements ............................................................................................. 12  
         2.2.6.5 Attributes ........................................................................................... 12  
         2.2.6.6 Groups ................................................................................................ 12  
         2.2.6.7 Attribute Groups ............................................................................... 12  
   2.2.6.7 XML Structures ......................................................................................... 11  

3 Protocol Details ....................................................................................................... 13  
   3.1 Common Details .................................................................................................. 13  
   3.2 Server Details ..................................................................................................... 13  
      3.2.1 Abstract Data Model .................................................................................. 13  
      3.2.2 Timers ......................................................................................................... 13  
      3.2.3 Initialization ................................................................................................ 13  
      3.2.4 Higher-Layer Triggered Events ................................................................. 13  
      3.2.5 Message Processing Events and Sequencing Rules ................................. 13  
         3.2.5.1 Admin_DeletePartition ....................................................................... 13  
         3.2.5.2 Admin_GetPartitionProperties .......................................................... 14  
         3.2.5.3 Admin_ListPartitions ......................................................................... 14  
         3.2.5.4 Admin_SetPartitionDataCacheVersion ............................................ 15  
         3.2.5.5 Admin_SetPartitionProperties ........................................................... 15  
         3.2.5.6 Admin_GetUpdatedPartitionProperties ........................................... 17  
         3.2.5.7 Admin_SetPartitionUserAcl ............................................................... 17  

[MS-UPASP2] — v20120630

Copyright © 2012 Microsoft Corporation.

Release: July 16, 2012
1 Introduction

The User Profile Admin Stored Procedure Protocol Specification enables the protocol client to interact with dedicated tenant-specific areas, called partitions, within a database that resides on the protocol server and contains properties that pertain to specific persons or entities.

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]:

- access control list (ACL)
- Coordinated Universal Time (UTC)
- GUID

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

- audience
- back-end database server
- canonical URL
- partition
- personal site
- Really Simple Syndication (RSS)
- request identifier
- result set
- return code
- stored procedure
- Structured Query Language (SQL)
- Transact-Structured Query Language (T-SQL)
- Uniform Resource Identifier (URI)
- Uniform Resource Locator (URL)
- user profile
- user profile store

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,
1.2.2 Informative References


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

1.3 Overview

This protocol provides a way for the protocol client to interact with parts (1) in the user profile store on the respective protocol server. It enables protocol clients to create, delete, or simply list parts (1) in the user profile store. It also provides a way for the protocol client to find out the number of user profiles, the number of audiences, and the number of parts that exist in the user profile store. In addition, it also allows protocol clients to read or update property values for a part (1) on the protocol server.

A protocol client may choose to implement a cache for property values associated with a part (1). In the case when a protocol client has such a caching mechanism, this protocol provides a way for maintaining a version number for the property values for a particular part (1) on the protocol server to facilitate cache invalidation on protocol client.

1.4 Relationship to Other Protocols
Figure 1: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

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

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

1.6 Applicability Statement

This protocol is designed to support a scale point of approximately:

- 1 partition (1) with 2 million user profiles
- 2 million partitions (1), each having 1 user profile
- Any other combination of number of partitions (1) and number of user profiles per partition (1), leading up to overall 2 million user profiles

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] specifies the transport protocol, Tabular Data Stream (TDS) Protocol, used to call the stored procedures, query SQL tables, get return codes, and return result sets mentioned in this document.

2.2 Common Data Types

2.2.1 Simple Data Types and Enumerations

No common simple data types or enumerations are defined in this protocol.

2.2.2 Bit Fields and Flag Structures

The bit fields and flag structures used in this specification are defined in this section.

2.2.2.1 User Rights Flags

An 8-byte unsigned long bit mask that specifies the user rights. The only valid values of the User Rights Flags bits are as follows.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>None</td>
</tr>
<tr>
<td>0x01</td>
<td>Manage personal site</td>
</tr>
<tr>
<td>0x02</td>
<td>Create personal site</td>
</tr>
<tr>
<td>0x04</td>
<td>Use social features</td>
</tr>
<tr>
<td>0x07</td>
<td>All</td>
</tr>
</tbody>
</table>

2.2.3 Binary Structures

No common binary structures are defined in this protocol.

2.2.4 Result Sets

2.2.4.1 Admin_ListPartitions.ResultSet0

The Admin_ListPartitions.ResultSet0 result set contains the list of all partition (1) identifiers that are present in the Tenants table.

<table>
<thead>
<tr>
<th>PartitionID uniqueidentifier,</th>
</tr>
</thead>
<tbody>
<tr>
<td>PartitionID: A GUID identifying a partition (1) in the user profile store. MUST be a value from column PartitionID of table Tenants.</td>
</tr>
</tbody>
</table>
2.2.4.2 PartitionProperties

The **PartitionProperties result set** contains information corresponding to partitions in the **Tenants** table.

```
PartitionID uniqueidentifier,
CanonicalMySitePortalUrl nvarchar(2084),
PreviousMySitePortalUrl nvarchar(2084),
CanonicalSearchCenterUrl nvarchar(2084),
PeopleResultsScope int,
DocumentResultsScope int,
DefaultRssFeed nvarchar(2084),
MySiteEmailSenderName nvarchar(max),
SynchronizationOU nvarchar(max),
ProfileMasterCacheVersion int,
DataCacheVersion int,
SerializedUserAcl nvarchar(max),
SecondaryMySiteOwner nvarchar(max),
NewsFeedEnabled bit,
MySiteMicroblogEMailsEnabled bit,
LangPacksApplied nvarchar(max),
```

**PartitionID**: A GUID identifying the partition (1) in the user profile store. MUST be a value from column PartitionID of table Tenants.

**CanonicalMySitePortalUrl**: A string specifying the canonical URL for the partition’s (1) my site portal. MUST be a value from column CanonicalMySitePortalUrl of table Tenants.

**PreviousMySitePortalUrl**: A string specifying the previous URL for the partition’s (1) my site portal. MUST be a value from column PreviousMySitePortalUrl of table Tenants.

**CanonicalSearchCenterUrl**: A string specifying the canonical URL for the partition’s (1) search center. MUST be a value from column CanonicalSearchCenterUrl of table Tenants.

**PeopleResultsScope**: An integer value specifying the scope for people search results. MUST be a value from column PeopleResultsScope of table Tenants.

**DocumentResultsScope**: An integer value specifying the scope for document search results. MUST be a value from column DocumentResultsScope of table Tenants.

**DefaultRssFeed**: A string specifying the default RSS feed URL. MUST be a value from column DefaultRssFeed of table Tenants.

**MySiteEmailSenderName**: A string specifying the sender e-mail address of my site e-mails. MUST be a value from column MySiteEmailSenderName of table Tenants.

**SynchronizationOU**: A string specifying the organizational unit of the partition (1) for synchronization. MUST be a value from column SynchronizationOU of table Tenants.

**ProfileMasterCacheVersion**: An integer value specifying the version number of the internal profile master cache. MUST receive a value from column ProfileMasterCacheVersion of table Tenants.

**DataCacheVersion**: An integer value specifying the version number of the internal data cache. MUST be a value from column DataCacheVersion of table Tenants.
SerializedUserAcl: A string specifying the serialized form of the user access control list (ACL) for the partition (1). MUST be a value from column SerializedUserAcl of table Tenants and MUST be a valid XML document instance of acl type defined in section 2.2.6.3.1.

SecondaryMySiteOwner: A string specifying the secondary my site owner. MUST be a value from column SecondaryMySiteOwner of table Tenants.

NewsFeedEnabled: A bit specifying whether or not the news feed is enabled. If the value is 1, the news feed MUST be enabled. If the value is 0, the news feed MUST be disabled. The value MUST NOT be NULL. MUST be a value from column NewsFeedEnabled of table Tenants.

MySiteMicroblogEMailsEnabled: A boolean value set to true enables the Microfeed feature to notify a user by email of their mention in a post, or when someone replies to the thread in which they are participating.

LangPacksApplied: A string specifying the language packs whose localizations have been applied to the default (or built-in or standard or whatever terminology makes sense) user profile properties. MUST be a value from the LangPacksApplied column of table Tenants.

2.2.5 Tables and Views

2.2.5.1 Tenants

The Tenants table stores information about the partitions (1) in the user profile store.

```
PartitionID uniqueidentifier NOT NULL,
LastModifiedTime datetime NOT NULL,
CanonicalMySitePortalUrl nvarchar(max) NOT NULL,
PreviousMySitePortalUrl nvarchar(max) NOT NULL,
CanonicalSearchCenterUrl nvarchar(max) NOT NULL,
PeopleResultsScope int NOT NULL,
DocumentResultsScope int NOT NULL,
DefaultRssFeed nvarchar(max) NOT NULL,
MySiteEmailSenderName nvarchar(max) NULL,
SynchronizationOU nvarchar(max) NULL,
ProfileMasterCacheVersion int NOT NULL,
SerializedUserAcl nvarchar(max) NULL,
DataCacheVersion int NOT NULL,
SecondaryMySiteOwner nvarchar(max) NULL,
NewsFeedEnabled bit NOT NULL,
LangPacksApplied nvarchar(max) NULL,
```

PartitionID: A GUID identifying the partition (1) in the user profile store. It is the @partitionID value supplied to proc_Admin_SetupPartitionedData stored procedure when the partition (1) was created.

LastModifiedTime: A time stamp in UTC format specifying the last time the partition properties were modified.

CanonicalMySitePortalUrl: A string specifying the canonical URL for the partition’s (1) my site portal.

PreviousMySitePortalUrl: A string specifying the previous URL for the partition’s (1) my site portal.
CanonicalSearchCenterUrl: A string specifying the canonical URL for the partition’s (1) search center.

PeopleResultsScope: An integer value specifying the scope for people (user profile) search results.

DocumentResultsScope: An integer value specifying the scope for document search results.

DefaultRssFeed: A string specifying the default RSS feed URL.

MySiteEmailSenderName: A string specifying the sender e-mail address of my site e-mails.

SynchronizationOU: A string specifying the organizational unit of the partition (1) for synchronization.

ProfileMasterCacheVersion: An integer value specifying the version number of the internal profile master cache.

SerializedUserAcl: A string value representing an XML document specifying the serialized form of the user access control list (ACL) for the partition (1). This MUST be a valid XML document instance of acl type defined in section 2.2.6.3.1.

DataCacheVersion: An integer value specifying the version number of the internal data cache.

SecondaryMySiteOwner: A string specifying the secondary my site owner.

NewsFeedEnabled: A bit value specifying whether or not the news feed is enabled.

LangPacksApplied: A string specifying the language packs whose localizations have been applied to the default (or built-in or standard or whatever terminology makes sense) user profile properties.

2.2.6 XML Structures

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

2.2.6.1 Namespaces

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Namespace URI</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td><a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a></td>
<td>[XMLSCHEMA1] [XMLSCHEMA2]</td>
</tr>
</tbody>
</table>

2.2.6.2 Simple Types

This specification does not define any common XML schema simple type definitions.

2.2.6.3 Complex Types

The following table summarizes the set of common XML schema complex type definitions defined by this specification.

<table>
<thead>
<tr>
<th>Complex Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>acl</td>
<td>Contains data about user access control list (ACL).</td>
</tr>
</tbody>
</table>
2.2.6.3.1 acl

The acl type contains data about user access control entries.

```xml
<xs:element name="acl">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ace" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:attribute name="identityName" type="xs:string" use="required"/>
          <xs:attribute name="displayName" type="xs:string" use="required"/>
          <xs:attribute name="sid" type="xs:string" use="required"/>
          <xs:attribute name="allowRights" type="xs:unsignedLong" use="required"/>
          <xs:attribute name="denyRights" type="xs:unsignedLong" use="required"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="version" type="xs:string" use="required"/>
  </xs:complexType>
</xs:element>
```

**ace**: Contains an access control entry.

**ace.identityName**: A string specifying the name of the identity for the access control entry.

**ace.displayName**: A string specifying the display name for the access control entry.

**ace.sid**: Base64 encoded string representation of security identifier bytes.

**ace.allowRights**: An unsigned long specifying the rights to be granted. This MUST be a value from User Rights Flags (section 2.2.2.1).

**ace.denyRights**: An unsigned long value specifying the rights to be denied. This MUST be a value from User Rights Flags (section 2.2.2.1).

**version**: A string specifying the version of the acl type.

2.2.6.4 Elements

This specification does not define any common XML schema element definitions.

2.2.6.5 Attributes

This specification does not define any common XML schema attribute definitions.

2.2.6.6 Groups

This specification does not define any common XML schema group definitions.

2.2.6.7 Attribute Groups

This specification does not define any common XML schema attribute group definitions.
3 Protocol Details

3.1 Common Details

None.

3.2 Server Details

The protocol server maintains information about partitions (1) in the user profile store and enables protocol clients to create, delete or simply list partitions (1) in the user profile store. In addition, it also allows protocol clients to read or update property values for a partition (1).

3.2.1 Abstract Data Model

In case of this protocol, protocol server is expected to have a user profile store. Users in the user profile store can be divided into multiple partitions such that user in a partition (1) does not have access to user profile data belonging to users in another partition (1).

Each partition (1) within user profile store has metadata associated with it stored in the Tenants table.

3.2.2 Timers

None.

3.2.3 Initialization

Authentication of the TDS connection to the server must occur before this protocol can be used. The data structures, stored procedures, and actual data are persisted by the protocol server within databases, any operation to initialize the state of the database MUST occur before the server can use this protocol.

3.2.4 Higher-Layer Triggered Events

None.

3.2.5 Message Processing Events and Sequencing Rules

3.2.5.1 Admin_DeletePartition

The Admin_DeletePartition stored procedure is called to remove the data of a specific partition (1) from the user profile store.

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

```sql
PROCEDURE Admin_DeletePartition (  
  @partitionID uniqueidentifier,
  @correlationId uniqueidentifier = null
);
```

@partitionID: A GUID identifying the partition (1) to be deleted.
@correlationId: The optional request identifier for the current request.
Return Values: An integer which MUST be in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The specified partition (1) does not exist.</td>
</tr>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
</tbody>
</table>

Result Sets: MUST NOT return any result sets.

### 3.2.5.2 Admin_GetPartitionProperties

The `Admin_GetPartitionProperties` stored procedure is called to get the properties for partitions (1). The T-SQL syntax for the stored procedure is as follows:

```sql
PROCEDURE Admin_GetPartitionProperties (   @top int = 1000   ,@lastPartitionID uniqueidentifier = null   ,@currentCachedTime datetime = null OUTPUT   ,@correlationId uniqueidentifier = null ) ;
```

- **@top**: An integer value specifying the maximum number of items in the returned result set. The value MUST be greater than 0.
- **@lastPartitionID**: A GUID value representing a partition (1) identifier that is used for comparison in selecting the items in the returned result set. If the value is NULL, first @top items are returned in the result set from the Tenants table sorted in ascending PartitionID order. If the value is NOT NULL then the first @top items whose PartitionID is greater than the provided value are returned in the result set from the Tenants table sorted in ascending PartitionID order.
- **@currentCachedTime**: A time stamp value in UTC format specifying the current time of the back-end database server. Any input value MUST be ignored.
- **@correlationId**: The optional request identifier for the current request.

Return Values: An integer which MUST be 0.

Result Sets:

This stored procedure MUST return a `PartitionProperties`.

### 3.2.5.3 Admin_ListPartitions

The `Admin_ListPartitions` stored procedure is called to retrieve the list of all partition (1) identifiers that are present in the Tenants table.

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

```sql
PROCEDURE Admin_ListPartitions ( ) ;
```

Return Values: An integer which MUST be 0.

Result Sets:
This stored procedure MUST return a `Admin_ListPartitions.ResultSet0`

### 3.2.5.4 Admin_SetPartitionDataCacheVersion

The `Admin_SetPartitionDataCacheVersion` stored procedure is called to set the version of the data cache of a specific partition (1). If the version value in column `DataCacheVersion` in the `Tenants` table matches the value in `@oldDataCacheVersion`, then the value in column `DataCacheVersion` is updated to the value specified in `@newDataCacheVersion` and the value in column `LastModifiedTime` is updated with the current time of the back-end database server in UTC format. Otherwise the values in `DataCacheVersion` and `LastModifiedTime` columns are not updated. In both cases, final value in the `DataCacheVersion` column is returned in `@finalDataCacheVersion`.

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

```sql
PROCEDURE Admin_SetPartitionDataCacheVersion (
    @partitionID uniqueidentifier,
    @oldDataCacheVersion int,
    @newDataCacheVersion int,
    @finalDataCacheVersion int OUTPUT,
    @correlationId uniqueidentifier = null);
```

- **@partitionID**: A GUID identifying the partition (1) for which the `DataCacheVersion` will be set. MUST be a value from column `PartitionID` of table `Tenants`.

- **@oldDataCacheVersion**: The old version of the data cache that the caller specifies. If the version value in column `DataCacheVersion` for the specified partition (1) in `Tenants` table matches the value in `@oldDataCacheVersion`, then the value in column `DataCacheVersion` MUST be updated to the value specified in `@newDataCacheVersion`. If the version value in column `DataCacheVersion` for the specified partition (1) in `Tenants` table does not match the value in `@oldDataCacheVersion`, then the value in column `DataCacheVersion` MUST NOT be updated.

- **@newDataCacheVersion**: The new version for the data cache.

- **@finalDataCacheVersion**: Output parameter which will receive the final version of the data cache from `DataCacheVersion` column in `Tenants` table for the specified partition (1).

- **@correlationId**: The optional request identifier for the current request.

**Return Values**: An integer which MUST be 0.

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

### 3.2.5.5 Admin_SetPartitionProperties

The `Admin_SetPartitionProperties` is called to set the properties for a partition (1).

Values in the `Tenants` table corresponding to the input parameters MUST be updated if the input parameter value is NOT NULL. The value in the `LastModifiedTime` column in the `Tenants` table MUST always be updated with the current time of the back-end database server in UTC format.

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

```sql
PROCEDURE Admin_SetPartitionProperties (
    @partitionID uniqueidentifier,
    @canonicalMySitePortalUrl nvarchar(2084) = null
);
```
@partitionID: A GUID identifying the partition (1) whose properties will be set. MUST be a value from column PartitionID of table Tenants.

@canonicalMySitePortalUrl: A string specifying the new canonical URL for the partition’s (1) my site portal. If this value is NULL, then the protocol server MUST NOT change the value in the user profile store.

@previousMySitePortalUrl: A string specifying the new value for previous URL for the partition’s (1) my site portal. If this value is NULL, then the protocol server MUST NOT change the value in the user profile store.

@canonicalSearchCenterUrl: A string specifying the new canonical URL for the partition’s (1) search center. If this value is NULL, then the protocol server MUST NOT change the value in the user profile store.

@peopleResultsScope: An integer value specifying the new scope for people search results. If this value is NULL, then the protocol server MUST NOT change the value in the user profile store.

@documentResultsScope: An integer value specifying the new scope for document search results. If this value is NULL, then the protocol server MUST NOT change the value in the user profile store.

@defaultRssFeed: A string specifying the new default RSS feed URL. If this value is NULL, then the protocol server MUST NOT change the value in the user profile store.

@mySiteEmailSenderName: A string specifying the new sender e-mail address of my site e-mails. If this value is NULL, then the protocol server MUST NOT change the value in the user profile store.

@synchronizationOU: A string specifying the new organizational unit of the partition (1) for synchronization. If this value is NULL, then the protocol server MUST NOT change the value in the user profile store.

@serializedUserAcl: A string specifying the new serialized form of the user access control list (ACL) for the partition (1). This value MUST be a valid XML document instance of acl type defined in section 2.2.6.3.1. If this value is NULL, then the protocol server MUST NOT change the value in the user profile store.

@profileMasterCacheVersion: An integer value specifying the version number of the internal profile master cache. If this value is NULL, then the protocol server MUST NOT change the value in the user profile store.

@secondaryMySiteOwner: A string specifying the new secondary my site owner. If this value is NULL, then the protocol server MUST NOT change the value in the user profile store.
@newsFeedEnabled: A bit value specifying whether or not the news feed is enabled. If this value is NULL, then the protocol server MUST NOT change the value in the user profile store.

@langPacksApplied: A string specifying the language packs whose localizations have been applied to the default (or built-in or standard or whatever terminology makes sense) user profile properties. If this value is NULL, then the protocol server MUST NOT change the value in the user profile store.

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

Return Values: An integer which MUST be 0.

Result Sets: MUST NOT return any result sets.

3.2.5.6 Admin_GetUpdatedPartitionProperties

The Admin_GetUpdatedPartitionProperties stored procedure is called to get the properties for partitions (1) that have been updated after a specific time.

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

```sql
PROCEDURE Admin_GetUpdatedPartitionProperties (  
  @lastCachedTime datetime  
  ,@currentCachedTime datetime = null OUTPUT  
  ,@correlationId uniqueidentifier = null
);
```

@lastCachedTime: A time stamp value in UTC format specifying the last time the partition (1) properties were queried by the caller. The result set MUST contain information corresponding to partitions that have a value in the LastModifiedTime column that is greater than lastCachedTime.

@currentCachedTime: A time stamp value in UTC format specifying the current time of the back-end database server. Any input value MUST be ignored.

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

Return Values: An integer that MUST be 0.

Result Sets:

This stored procedure MUST return a PartitionProperties

3.2.5.7 Admin_SetPartitionUserAcl

The Admin_SetPartitionUserAcl is called to set the serialized form of the user access control list (ACL) for the partition (1).

If the ACL value in column SerializedUserAcl in Tenants table matches the value in @oldSerializedUserAcl, then the value in columnSerializedUserAcl is updated to the value specified in @newSerializedUserAcl, the value in column LastModifiedTime is updated with the current time of the back-end database server in UTC format and 0 is returned to indicate successful execution. Otherwise the values in SerializedUserAcl and LastModifiedTime columns are not updated and 1 is returned to indicate concurrent update error.

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

```sql
PROCEDURE Admin_SetPartitionUserAcl (  
```

[MS-UPASP2] — v20120630

Copyright © 2012 Microsoft Corporation.

Release: July 16, 2012
@partitionID uniqueidentifier
,@oldSerializedUserAcl nvarchar(max)
,@newSerializedUserAcl nvarchar(max)
,@correlationId uniqueidentifier = null
);

@partitionID: A GUID identifying the partition (1) whose ACL property will be set. MUST be a value from column PartitionID of table Tenants.

@oldSerializedUserAcl: A string specifying the latest value for the serialized form of the user access control list (ACL) for the partition (1) which the caller retrieved from the user profile store. This value MUST be a valid XML document instance of acl type defined in section 2.2.6.3.1.

@newSerializedUserAcl: A string specifying the new value for the serialized form of the user access control list (ACL) for the partition (1). This value MUST be a valid XML document instance of acl type defined in section 2.2.6.3.1.

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

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

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Concurrent update error.</td>
</tr>
<tr>
<td>0</td>
<td>Successful execution.</td>
</tr>
</tbody>
</table>

Result Sets: MUST NOT return any result sets.

3.2.5.8 proc_Admin_DeletePartitionedData

Delete a partition’s data for a given object. The procedure MUST be called for every partitioned data object within a single transaction.

PROCEDURE proc_Admin_DeletePartitionedData (  
@partitionID uniqueidentifier  
,@tableName nvarchar(517)  
,@correlationId uniqueidentifier = null  
);

@partitionID: A GUID identifying the partition (1) whose object data will be deleted. MUST be a value from column PartitionID of table Tenants.

@tableName: A data object name from the TableName column of result set of proc_Admin_ListPartitionedTables.

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

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

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>The procedure was not called in a transaction or @tableName is not the name of a partitioned data object.</td>
</tr>
</tbody>
</table>
### Value | Description
---|---
0 | Success

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

#### 3.2.5.9 proc_Admin_SetupPartitionedData

Add a new partition's data for a given object. The procedure MUST be called for every partitioned data object within a single transaction.

```sql
PROCEDURE proc_Admin_SetupPartitionedData (
    @partitionID uniqueidentifier,
    @tableName nvarchar(517),
    @correlationId uniqueidentifier = null
);
```

- **@partitionID:** A GUID identifying the partition (1) whose object data will be added. MUST be a value from column PartitionID of table `Tenants`.
- **@tableName:** A data object name from the TableName column of result set of proc_Admin_ListPartitionedTables.
- **@correlationId:** The optional request identifier for the current request.

**Return Values:** An integer which MUST be in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>The procedure was not called in a transaction or @tableName is not the name of a partitioned data object.</td>
</tr>
<tr>
<td>0</td>
<td>Success.</td>
</tr>
</tbody>
</table>

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

#### 3.2.6 Timer Events

None.

#### 3.2.7 Other Local Events

None.

#### 3.3 Client Details

None.

#### 3.3.1 Abstract Data Model

None.
3.3.2 Timers  
None.

3.3.3 Initialization  
None.

3.3.4 Higher-Layer Triggered Events  
None.

3.3.5 Message Processing Events and Sequencing Rules  
None.

3.3.6 Timer Events  
None.

3.3.7 Other Local Events  
None.
4 Protocol Examples

4.1 Create a New Partition

This example describes the requests that are made when a user profile store administrator creates
and sets up a new partition (1). This example assumes that the user profile store was already
created with a default partition (1).

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

1. -- proc_Admin_SetupPartitionedData -->
2. <-- Return code is ignored --
3. -- Admin_SetPartitionProperties -->
4. <-- Return code is ignored --

The protocol client first calls proc_Admin_SetupPartitionedData to create the partition:

exec dbo. proc_Admin_SetupPartitionedData
@partitionID='7A9E3CAC-0B81-49A0-BFEE-5C33A3874916',
@correlationId='00000000-0000-0000-0000-000000000000'

The protocol server creates a partition from this information and returns 1, which is ignored.

The protocol client then calls Admin_SetPartitionProperties

exec dbo.Admin_SetPartitionProperties
@partitionID='7A9E3CAC-0B81-49A0-BFEE-5C33A3874916',
@correlationId='00000000-0000-0000-0000-000000000000',
@canonicalMySitePortalUrl=N'http://server.example.com/My/',
@canonicalSearchCenterUrl=N'',
@peopleResultsScope=0,
@documentResultsScope=1,
@defaultRssFeed=N'',
@mySiteEmailSenderName=N'MySite',
@synchronizationOU=default,
serializedUserAcl=N'<acl version="1.0"><ace identityName="NT AUTHORITY\authenticated users" displayName="NT AUTHORITY\Authenticated Users" sid="AQEAAAAAAAULAAAA" allowRights="7" denyRights="0" /></acl>'
profileMasterCacheVersion=0

The protocol server sets up the partition according to these parameters and then returns 0, which is
ignored.

4.2 Verify Partition Information

In this example, a user profile store administrator wants to verify information about the partition (1)
they created in the previous example.

The protocol client then calls Admin_ListPartitions.

exec dbo.Admin_ListPartitions
The protocol server responds with the following results.

<table>
<thead>
<tr>
<th>PartitionID</th>
</tr>
</thead>
<tbody>
<tr>
<td>0C37852B-34D0-418E-91C6-2AC25AF4BE5B</td>
</tr>
<tr>
<td>7A9E3CAC-0B81-49A0-BFEE-5C33A3874916</td>
</tr>
</tbody>
</table>

Using the PartitionID which matches the identifier that the administrator passed earlier, the protocol client now calls Admin_GetPartitionProperties.

```
DECLARE @p4 DATETIME
SET @p4='2010-01-15 17:51:09.6000000'
EXEC dbo.Admin_GetPartitionProperties
@correlationId='806597C7-2A34-4BB3-A807-A8664115E8D1',
@top=1000,
@lastPartitionID=NULL,
@currentCachedTime=@p4 output
SELECT @p4
```

The protocol server gets the properties requested and responds with the following output. It also returns result set, detailed following, and a return value of 0 which is ignored. Several columns in the result set which are empty have been omitted for clarity (omitted columns are PreviousMySitePortalUrl, CanonicalSearchCenterUrl, SynchronizationOU and DefaultRssFeed).

```
currentCachedTime
2010-01-15 17:52:15.917
```

<table>
<thead>
<tr>
<th>Partition Identifier</th>
<th>Canonical MySite PortalUrl</th>
<th>People Result Scope</th>
<th>Document Result Scope</th>
<th>MySite Email Sender Name</th>
<th>Profile Master Cache Version</th>
<th>Data Cache Version</th>
<th>Serialized UserACL</th>
</tr>
</thead>
<tbody>
<tr>
<td>0C37852B-34D0-418E-91C6-2AC25AF4BE5B</td>
<td><a href="http://server.example.com/host/My/">http://server.example.com/host/My/</a></td>
<td>0</td>
<td>1</td>
<td>MySite</td>
<td>0</td>
<td>1</td>
<td>&lt;acl version=&quot;1.0&quot;&gt; &lt;ace identityName=&quot;nt auth authority\authenticated users&quot; displayName=&quot;NT AUTHORITY\Authenticated Users&quot; sid=&quot;AQEAAAAAAA UAAAA&quot; allowRights=&quot;7&quot; denyRights=&quot;0&quot; /&gt;&lt;/ace&gt;</td>
</tr>
<tr>
<td>7A9E3CAC-0B81-49A0-BFEE-5C33A387</td>
<td><a href="http://server.example.com/My/">http://server.example.com/My/</a></td>
<td>0</td>
<td>1</td>
<td>MySite</td>
<td>0</td>
<td>1</td>
<td>&lt;acl version=&quot;1.0&quot;&gt; &lt;ace identityName=&quot;nt auth authority\authenticated users&quot; displayName=&quot;NT AUTHORITY\Authenticated Users&quot; sid=&quot;AQEAAAAAAA UAAAA&quot; allowRights=&quot;7&quot; denyRights=&quot;0&quot; /&gt;&lt;/ace&gt;</td>
</tr>
<tr>
<td>Partition Identifier</td>
<td>Canonical MySite PortalUrl</td>
<td>People Results Scope</td>
<td>Document Results Scope</td>
<td>MySite Email Sender Name</td>
<td>Profile Master Cache Version</td>
<td>Data Cache Version</td>
<td>Serialized UserACL</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------------------------</td>
<td>----------------------</td>
<td>------------------------</td>
<td>--------------------------</td>
<td>-----------------------------</td>
<td>-------------------</td>
<td>-------------------</td>
</tr>
</tbody>
</table>
| 4916                 |                           |                      |                        |                          |                             |                   | <acl>

  
  ated users" displayName="NT AUTHORITY\Authenticated Users"
  sid="AQEAAAAAAA ULAAAA"
  allowRights="7"
  denyRights="0"
</acl>
5 Security

5.1 Security Considerations for Implementers

Interactions with SQL are susceptible to tampering and other types of security risks. Implementers are advised to sanitize the input parameters for a stored procedure before invoking the stored procedure.

5.2 Index of Security Parameters

None.
6 Appendix A: Product Behavior

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

- Microsoft® SharePoint® Server 2013 Preview
- Microsoft® SQL Server® 2008 R2 SP1
- Microsoft® SQL Server® 2012

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 13

acl - complex type 12
Admin_DeletePartition method 13
Admin_GetPartitionProperties method 14
Admin_GetUpdatedPartitionProperties method 17
Admin_ListPartitions method 14
Admin_ListPartitions.ResultSet0 result set 8
Admin_SetPartitionDataCacheVersion method 15
Admin_SetPartitionProperties method 15
Admin_SetPartitionUserAcl method 17
Applicability 7
Attribute groups - overview 12
Attributes - overview 12

B

Binary structures - overview 8
Bit fields - overview 8

C

Capability negotiation 7
Change tracking 26
Client
  
  abstract data model 19
  
  higher-layer triggered events 20
  
  initialization 20
  
  local events 20
  
  message processing 20
  
  overview 19
  
  sequencing rules 20
  
  timer events 20
  
  timers 20

Complex types
  
  acl 12

Complex types - overview 11
Create a new partition example 21

D

Data model - abstract
  
  client 19
  
  server 13

Data types - simple
  
  overview 8

E

Elements - overview 12
Events
  
  local - client 20
  
  local - server 19
  
  timer - client 20
  
  timer - server 19

Examples

  create a new partition 21
  
  verify partition information 21

F

Fields - vendor-extensible 7
Flag structures - overview 8
Flags - messages
  
  user rights 8

G

Glossary 5
Groups - overview 12

H

Higher-layer triggered events
  
  client 20
  
  server 13

I

Implementer - security considerations 24
Index of security parameters 24
Informative references 6
Initialization
  
  client 20
  
  server 13

Introduction 5

L

Local events
  
  client 20
  
  server 19

M

Message processing
  
  client 20

Messages
  
  acl complex type 12
  
  Admin_ListPartitions.ResultSet0 result set 8
  
  attribute groups 12
  
  attributes 12
  
  binary structures 8
  
  bit fields 8
  
  complex types 11
  
  elements 12
  
  enumerations 8
  
  flag structures 8
  
  groups 12
  
  namespaces 11
  
  PartitionProperties result set 9
  
  simple data types 8
  
  simple types 11
  
  tenants table structure 10
  
  transport 8

[MS-UPASP2] — v20120630

Copyright © 2012 Microsoft Corporation.

Release: July 16, 2012