Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft [Open Specification Promise](#) or the [Community Promise](#). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.

- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.

- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

**Tools.** The Open Specifications do not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them. Certain Open Specifications are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.
## Revision Summary

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision History</th>
<th>Revision Class</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>04/04/2008</td>
<td>0.1</td>
<td></td>
<td>Initial Availability</td>
</tr>
<tr>
<td>06/27/2008</td>
<td>1.0</td>
<td>Major</td>
<td>Revised and edited the technical content</td>
</tr>
<tr>
<td>12/12/2008</td>
<td>1.01</td>
<td>Editorial</td>
<td>Revised and edited the technical content</td>
</tr>
<tr>
<td>03/18/2009</td>
<td>1.02</td>
<td>Editorial</td>
<td>Revised and edited the technical content</td>
</tr>
<tr>
<td>07/13/2009</td>
<td>1.03</td>
<td>Major</td>
<td>Changes made for template compliance</td>
</tr>
<tr>
<td>08/28/2009</td>
<td>1.04</td>
<td>Editorial</td>
<td>Revised and edited the technical content</td>
</tr>
<tr>
<td>11/06/2009</td>
<td>1.05</td>
<td>Editorial</td>
<td>Revised and edited the technical content</td>
</tr>
<tr>
<td>02/19/2010</td>
<td>2.0</td>
<td>Editorial</td>
<td>Revised and edited the technical content</td>
</tr>
<tr>
<td>03/31/2010</td>
<td>2.01</td>
<td>Editorial</td>
<td>Revised and edited the technical content</td>
</tr>
<tr>
<td>04/30/2010</td>
<td>2.02</td>
<td>Editorial</td>
<td>Revised and edited the technical content</td>
</tr>
<tr>
<td>06/07/2010</td>
<td>2.03</td>
<td>Editorial</td>
<td>Revised and edited the technical content</td>
</tr>
<tr>
<td>06/29/2010</td>
<td>2.04</td>
<td>Editorial</td>
<td>Changed language and formatting in the technical content.</td>
</tr>
<tr>
<td>07/23/2010</td>
<td>2.04</td>
<td>No change</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>09/27/2010</td>
<td>2.04</td>
<td>No change</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>11/15/2010</td>
<td>2.04</td>
<td>No change</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>12/17/2010</td>
<td>2.04</td>
<td>No change</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>03/18/2011</td>
<td>2.04</td>
<td>No change</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>06/10/2011</td>
<td>2.04</td>
<td>No change</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>01/20/2012</td>
<td>2.5</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>04/11/2012</td>
<td>2.5</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>2.5</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 .......................................................................................................................... 6  
   1.2.1 Normative References .................................................................................................. 6  
   1.2.2 Informative References ............................................................................................... 6  
1.3 Protocol Overview (Synopsis) ............................................................................................. 6  
1.4 Relationship to Other Protocols .......................................................................................... 7  
1.5 Prerequisites/Preconditions ............................................................................................... 8  
1.6 Applicability Statement ...................................................................................................... 8  
1.7 Versioning and Capability Negotiation ................................................................................. 8  
1.8 Vendor-Extensible Fields .................................................................................................... 8  
1.9 Standards Assignments ....................................................................................................... 8  

2 **Messages** .......................................................................................................................... 9  
2.1 Transport .............................................................................................................................. 9  
2.2 Common Data Types ............................................................................................................ 9  
   2.2.1 Simple Data Types and Enumerations ......................................................................... 9  
   2.2.2 Simple Data Types ....................................................................................................... 9  
      2.2.2.1 Privacy Policy Type ................................................................................................ 9  
      2.2.2.2 Change Type ......................................................................................................... 9  
      2.2.2.3 Object Type .......................................................................................................... 9  
      2.2.2.4 Value .................................................................................................................... 10  
   2.2.3 Bit Fields and Flag Structures ....................................................................................... 11  
   2.2.4 Binary Structures ....................................................................................................... 11  
   2.2.5 Result Sets .................................................................................................................. 11  
   2.2.6 Tables and Views ......................................................................................................... 11  
   2.2.7 XML Structures .......................................................................................................... 11  
      2.2.7.1 Namespaces ........................................................................................................... 11  
      2.2.7.2 Simple Types ......................................................................................................... 11  
      2.2.7.3 Complex Types .................................................................................................... 11  
      2.2.7.3.1 WebLogPostDescription ............................................................................... 11  
         2.2.7.3.1.1 WebLogPostDescription Schema ................................................................. 12  
      2.2.7.4 Attributes .............................................................................................................. 12  
      2.2.7.5 Elements ............................................................................................................... 12  
      2.2.7.6 Groups .................................................................................................................. 12  
      2.2.7.7 Attribute Groups ................................................................................................... 12  

3 **Protocol Details** ................................................................................................................. 13  
3.1 User Profile Change Log Server Details .......................................................................... 13  
3.1.1 Abstract Data Model ..................................................................................................... 13  
3.1.2 Timers ............................................................................................................................ 14  
3.1.3 Initialization .................................................................................................................. 14  
3.1.4 Higher-Layer Triggered Events ..................................................................................... 14  
3.1.5 Message Processing Events and Sequencing Rules ....................................................... 14  
   3.1.5.1 profile_AddWebLogEvent .................................................................................... 14  
   3.1.5.2 profile_RemoveWebLogEvent .............................................................................. 15  
   3.1.5.3 profile_GetUserEvents ......................................................................................... 15  
      3.1.5.3.1 profile_GetUserEvents_Last_Event Result Set ............................................... 17  
      3.1.5.3.2 profile_GetUserEvents_Event_Log_1 Result Set ........................................... 17  
      3.1.5.3.3 profile_GetUserEvents_Event_Log_2 Result Set ........................................... 18  

[MS-UPSCHNG] — v20120630  
User Profile Change Log Stored Procedure Protocol Specification  
Copyright © 2012 Microsoft Corporation.  
Release: July 16, 2012
1 Introduction

This document provides specific details of the User Profile Change Log Stored Procedure protocol. This protocol allows multiple ways for the protocol client to interact with the user profile change events on the protocol server.

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
- Security Support Provider Interface (SSPI)

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

- blog
- change log
- change token
- colleague
- datetime
- display name
- distribution list
- e-mail address
- event
- front-end Web server
- item
- membership
- multivalue property
- organization
- personal site
- result set
- return code
- Session Initiation Protocol (SIP) address
- single-value property
- stored procedure
- Structured Query Language (SQL)
- Transact-Structured Query Language (T-SQL)
- Uniform Resource Locator (URL)
- user profile
- user profile change event
- user profile privacy policy

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.


1.2.2 Informative References


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

1.3 Protocol Overview (Synopsis)

This protocol allows clients to add, or delete user profile change events from a log on the protocol server, as well as retrieve those user profile change events using pre-defined criteria such as events which have happened after a specified time or with a specific user.

The following diagram shows the data flow between the protocol client and the protocol server.
The protocol client can ask the protocol server to add or remove a blog post user profile change event, or the protocol client can ask the protocol server to generate all user profile change events with an Object Type of "Anniversary" (see section 2.2.2.3).

The protocol client can request the protocol server to provide all user profile change events for a particular user or for all of the colleague properties for a specified user. In another possible operation, the protocol client can request the protocol server to provide the change token for the last read user profile change event.

1.4 Relationship to Other Protocols

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

![Figure 1: User Profile Change Log protocol data flow](image)
1.5 Prerequisites/Preconditions

None.

1.6 Applicability Statement

This protocol is used to interact with a change log of user events on protocol server.

1.7 Versioning and Capability Negotiation

Versions of the data structures or stored procedures in the database need to be the same as expected by the front-end Web server. If the stored procedures do not provide the calling parameters or return values as expected, the results of the call are indeterminate.

The version negotiation process for this protocol is identical to the process defined in [MS-WSSFO], section 1.7.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.
2 Messages

2.1 Transport

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

2.2 Common Data Types

The following are common data types used in conjunction with this protocol.

2.2.1 Simple Data Types and Enumerations

2.2.2 Simple Data Types

2.2.2.1 Privacy Policy Type

A 32-bit signed integer. The only valid values of a user profile privacy policy type for a specified item are as follows.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The items that have this privacy policy setting are mandatory.</td>
</tr>
<tr>
<td>2</td>
<td>The items that have this privacy policy setting are enabled.</td>
</tr>
<tr>
<td>4</td>
<td>The items that have this privacy policy were opted to be disabled.</td>
</tr>
<tr>
<td>8</td>
<td>The items that have this privacy policy are disabled.</td>
</tr>
</tbody>
</table>

2.2.2.2 Change Type

A 32-bit signed integer. Change Type MUST be composed doing a BINARY OR of 1 or more values from the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Type</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Add</td>
<td>An object has been added to the user profile.</td>
</tr>
<tr>
<td>2</td>
<td>Modify</td>
<td>An existing object has been modified</td>
</tr>
<tr>
<td>4</td>
<td>Delete</td>
<td>An existing object has been deleted.</td>
</tr>
<tr>
<td>8</td>
<td>Metadata</td>
<td>The metadata of the object has been modified</td>
</tr>
</tbody>
</table>

2.2.2.3 Object Type

A 32-bit signed integer.

For a stored procedure input parameter, the Object Type value MUST be composed by using a binary OR of one or more values from the following table.

For stored procedure result sets, the Object Type value MUST be one and only one of the values in the following table.
### 2.2.2.4 Value

The value of a user profile change entry defined by the Object Type (Section 2.2.2.3) and MUST be one of the following representations:

<table>
<thead>
<tr>
<th>Object Type</th>
<th>Value Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anniversary</td>
<td>T-SQL nvarchar value containing the anniversary UTC date and time, and uses the</td>
</tr>
<tr>
<td></td>
<td>following format: &quot;yyyy-MM-dd HH:mm:ss&quot;</td>
</tr>
<tr>
<td>DLMembership</td>
<td>T-SQL nvarchar value that uniquely identifies the distribution list.</td>
</tr>
<tr>
<td>SiteMembership</td>
<td>T-SQL nvarchar value containing the site GUID</td>
</tr>
<tr>
<td>QuickLink</td>
<td>T-SQL nvarchar value containing the Uniform Resource Locator (URL) for the link.</td>
</tr>
<tr>
<td>Colleague</td>
<td>T-SQL nvarchar value containing the user name of the colleague.</td>
</tr>
<tr>
<td>PersonalizationSite</td>
<td>T-SQL nvarchar value containing a link for the specified personal site.</td>
</tr>
<tr>
<td>UserProfile</td>
<td>T-SQL nvarchar value containing the name of the user whose user profile was changed.</td>
</tr>
<tr>
<td>WebLog</td>
<td>MUST be NULL.</td>
</tr>
<tr>
<td>SingleValueProperty</td>
<td>Any of the following, depending on the property type:</td>
</tr>
<tr>
<td>or MultiValueProperty</td>
<td>- Boolean value represented as a string containing 0 or 1.</td>
</tr>
<tr>
<td></td>
<td>- T-SQL nvarchar value.</td>
</tr>
<tr>
<td></td>
<td>- UTC Date – encoded as &quot;yyyy-MM-dd HH:mm:ss&quot;.</td>
</tr>
<tr>
<td></td>
<td>- Float – in single precision SQL-defined floating point number encoded as a string</td>
</tr>
<tr>
<td></td>
<td>in the current culture.</td>
</tr>
<tr>
<td>Object Type</td>
<td>Value Representation</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------------</td>
</tr>
<tr>
<td></td>
<td>int32 – 32 bit signed integer.</td>
</tr>
<tr>
<td></td>
<td>int64 – 64 bit signed integer.</td>
</tr>
</tbody>
</table>

2.2.3 Bit Fields and Flag Structures

None.

2.2.4 Binary Structures

None.

2.2.5 Result Sets

None.

2.2.6 Tables and Views

None.

2.2.7 XML Structures

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

2.2.7.1 Namespaces

None.

2.2.7.2 Simple Types

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

2.2.7.3 Complex Types

XML Schema complex type definitions that are specific to a particular operation are described with each operation.

2.2.7.3.1 WebLogPostDescription

The WebLogPostDescription type contains data about a blog post as specified as follows.

Usage Scenario

```
<WebLog>
  <Title>value</Title>
  <Permalink>value</Permalink>
</WebLog>
```
2.2.7.3.1.1 WebLogPostDescription Schema

```xml
<s:element name="WebLog">
  <s:complexType>
    <s:sequence>
      <s:element name="Title" type="s:string" />
      <s:element name="Permalink" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
```

**Title:** This element specifies the name of the blog post as a string.

**Permalink:** This element specifies a URL which can be used to access the blog.

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 User Profile Change Log Server Details

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.

Figure 3: Abstract data model

In the preceding diagram, each table specifies a type of entity in the model, and each line specifies that one type of entity always contains a reference to another.

- **UserProfile Table**: A collection of entries corresponding to the user profile available on the server.
  - **UserID**: A unique identifier for a user.

- **Colleagues Table**: A collection of entries, where each one corresponds to a colleague of a user.
  - **UserID**: The unique identifier of the user.
  - **ColleagueID**: The unique identifier of the user colleague.

- **PropertyList Table**: A collection of entries, each one corresponding to a property.
  - **PropertyName**: A unique name for the property.
  - **PropertyType**: An integer identifying the type of the property.
  - **State**: An integer identifying what is the property state.

- **PropertyValues Table**: A collection of entries with the values of the user properties.
- **UserID**: The unique identifier of the user.
- **PropertyName**: The unique name of the property.
- **Value**: The value of the property identified by **PropertyName** for the user identified by **UserID**.
- **Privacy**: An integer indicating what is the privacy policy for this property value.

**UserProfileChangeLog**: A collection of entries containing the user profile change entries.

- **EventID**: A unique identifier for the user profile change log entry.
- **UserID**: The unique identifier for the user for whom the user profile change log entry was created.
- **ChangeType**: An integer identifying the Change Type of the user profile change log entry.
- **ObjectType**: An integer identifying the Object Type of the user profile change log entry.
- **PropertyName**: The name of the single-value property or multivalue property which the user profile change log entry refers to.
- **Value**: The value of the property for the user profile change log entry.

**ChangeTime**: The Date and Time the user profile change log entry was created.

### 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

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

Any column in the result sets that does not have a defined name in the current implementation is designated curly braces in the form \{name\}. This does not affect the operation of the result set, as the ordinal position of any column with no defined name is expected by the front-end Web server.

#### 3.1.5.1 profile_AddWebLogEvent

The `profile_AddWebLogEvent` stored procedure is called to create a user profile change event describing an added blog post. The created user profile change event MUST have a user profile Change Type of "Add" and an Object Type of "WebLog".
profile_AddWebLogEvent is defined using T-SQL syntax as follows:

```sql
PROCEDURE profile_AddWebLogEvent(
    @RecordId bigint,
    @Data sql_variant
);
```

@RecordId: The record identifier that identifies the user who added the blog post. This parameter MUST not be NULL.

@Data: A WebLogPostDescription (Section 2.2.7.3.1) which specifies the blog post corresponding to the user profile change event to be created. This parameter MUST be specified.

Return Code Values: profile_AddWebLogEvent MUST return 0.

Result Sets: profile_AddWebLogEvent MUST NOT return any result set.

3.1.5.2 profile_RemoveWebLogEvent

The profile_RemoveWebLogEvent stored procedure is called to remove a user profile change event that has a user profile Change Type of "Add" and an Object Type of "WebLog".

profile_RemoveWebLogEvent is defined using T-SQL syntax as follows:

```sql
PROCEDURE profile_RemoveWebLogEvent (
    @RecordId bigint,
    @Data sql_variant
);
```

@RecordId: The Record Identifier that identifies the user who created the blog post. This parameter MUST not be NULL.

@Data: A WebLogPostDescription (Section 2.2.7.3.1) specifying the blog post corresponding to the user profile change event to be removed. This parameter MUST be specified.

Return Code Values: profile_RemoveWebLogEvent MUST return 0.

Result Sets: profile_RemoveWebLogEvent MUST NOT return any result set.

3.1.5.3 profile_GetUserEvents

The profile_GetUserEvents stored procedure is called to retrieve user profile change events.

profile_GetUserEvents is defined using T-SQL syntax as follows:

```sql
PROCEDURE profile_GetUserEvents ( 
    @RecordId bigint = NULL,
    @ViewerRights int,
    @MinEventId bigint = NULL,
    @MinEventTime datetime = NULL,
    @ChangeTypeMask int,
    @ObjectTypeMask int
);
```
**@RecordId**: A record identifier of a user, or NULL. If @RecordId is specified, `profile_GetUserEvents` MUST return the user profile change events for the user identified. If @RecordId is NULL, then `profile_GetUserEvents` MUST return all available user profile change events for existing users.

**@ViewerRights**: A bitmask specifying the privacy policy type that MUST be satisfied by the returned user profile change events. Each bit corresponds to a privacy level. If @RecordId is not NULL, then this value MUST be specified.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>All users are allowed to access the resource.</td>
</tr>
<tr>
<td>2</td>
<td>The only users allowed to access the resource are the owner of the resource and the owner’s colleagues.</td>
</tr>
<tr>
<td>4</td>
<td>The only users allowed to access the resource are the owner of the resource and the owner’s workgroup colleagues.</td>
</tr>
<tr>
<td>8</td>
<td>The only two users allowed to access the resource are the owner of the resource and the manager of the owner.</td>
</tr>
<tr>
<td>16</td>
<td>The only user allowed to access the resource is the owner of the resource.</td>
</tr>
</tbody>
</table>

For example, 0x0C signifies the Manager and organization Privacy Policy Type values. This parameter SHOULD be at least 0x01 (Public).

**@MinEventId**: A unique identifier for user profile change events. If @MinEventId is not NULL then all user profile change events with identifiers greater than this value MUST be returned. If @MinEventId is NULL then @MinEventTime MUST NOT be NULL.

**@MinEventTime**: A value representing the date and time for which all returned user profile change events MUST be more recent. If NULL then it MUST return all available user profile change events. If @MinEventId is NULL then @MinEventTime MUST NOT be NULL. If @MinEventId is specified then @MinEventTime MUST be ignored.

**@ChangeTypeMask**: A Change Type value that specifies the user profile Change Types of the user profile change events that MUST be returned.

**@ObjectTypeMask**: An Object Type value that specifies the Object Type of the user profile change events that MUST be returned.

**Return Code Values**: `profile_GetUserEvents` MUST return 0.

**Result Sets**: `profile_GetUserEvents` MUST return two of the following three result sets:

- `profile_GetUserEvents_LastEvent`
- `profile_GetUserEvents_Event_Log_1`
- `profile_GetUserEvents_Event_Log_2`

If @RecordId is NULL then `profile_GetUserEvents` MUST return the `profile_GetUserEvents_LastEvent` and `profile_GetUserEvents_Event_Log_1` result sets.

If @RecordId is not NULL then `profile_GetUserEvents` MUST return the `profile_GetUserEvents_LastEvent` and `profile_GetUserEvents_Event_Log_2` result sets.
3.1.5.3.1 profile_GetUserEvents_Last_Event Result Set

The profile_GetUserEvents_Last_Event result set MUST contain one record identifying the most recent user profile change event prior to the specified @MinEventTime or @MinEventId parameter. The profile_GetUserEvents_Last_Event result set is defined using T-SQL syntax as follows:

```sql
EventTime   datetime,
EventId     bigint;
```

**EventTime:** The date and time when the user profile change event occurred.

**EventId:** Unique identifier for the user profile change event.

3.1.5.3.2 profile_GetUserEvents_Event_Log_1 Result Set

The profile_GetUserEvents_Event_Log_1 result set returns the user profile change events for all existing users up to a maximum of 1000 records ordered from the earliest to the most recent. The profile_GetUserEvents_Event_Log_1 result set is defined using T-SQL syntax as follows:

```sql
EventId     bigint,
RecordId    bigint,
ChangeType  int,
EventTime   datetime,
OldValue    bigint,
NewValueData sql_variant,
NewValueChecksum int,
ObjectType  int,
ItemSecurity int,
ChangedLinkId bigint,
ChangedColleagueId bigint,
ChangedMemberGroupId bigint,
ChangedPropertyId bigint,
ChangedSourceId uniqueidentifier,
UserId      uniqueidentifier,
NTName      nvarchar(400),
Email       nvarchar(256),
SipAddress  nvarchar(250),
PreferredName nvarchar(256);
```

**EventId:** Unique identifier for the user profile change event.

**RecordId:** Identifier for the user that created the user profile change event.

**ChangeType:** Numeric value representing the user profile Change Type of the user profile event.

**EventTime:** Date and time when the user profile change event occurred.

**OldValue:** This value MUST be NULL.

**NewValueData:** The current value.

**NewValueChecksum:** Numeric value as returned by the T-SQL CHECKSUM function of the actual value.
**ObjectType:** The Object Type of the user profile event.

**ItemSecurity:** The Privacy Policy Type value. This value MUST NOT be NULL if the Object Type is either of the values QuickLink or PersonalizationSite.

**ChangedLinkId:** Unique identifier for the changed link. This value MUST NOT be NULL if the Object Type is either of the values QuickLink or PersonalizationSite.

**ChangedColleagueId:** The record identifier of the Colleague property that has changed. This value MUST NOT be NULL if the Object Type is the value Colleague.

**ChangedMemberGroupId:** The record identifier of a Member Group which corresponds to the user's membership which changed. This value MUST NOT be NULL if the Object Type is either of the values DLMembership or SiteMembership.

**ChangedPropertyId:** The unique identifier for user profile property that has changed. This value MUST NOT be NULL if the Object Type is either of the values SingleValueProperty or MultiValueProperty.

**ChangedSourceId:** GUID for the Privacy Policy record. This value MUST NOT be NULL for all Object Type values, with the exception of WebLog and UserProfile.

**UserId:** Record identifier for the user with whom the event is associated. This value MUST be NULL if @RecordId is not NULL.

**NTName:** User name for the user with whom the event is associated. This value MUST be NULL if @RecordId is not NULL.

**Email:** An e-mail address for the user with whom the event is associated. This value MUST be NULL if @RecordId is not NULL.

**SipAddress:** The Session Initiation Protocol (SIP) address for the user with whom the event is associated. This value MUST be NULL if @RecordId is not NULL.

**PreferredName:** Display name for the user with whom the event is associated. This value MUST be NULL if @RecordId is not NULL.

### 3.1.5.3.3 profile_GetUserEvents_Event_Log_2 Result Set

The profile_GetUserEvents_Event_Log_2 result set returns the user profile change events for the specified user, up to a maximum of 1000 records ordered from the earliest to the latest.

The profile_GetUserEvents_Event_Log_2 result set is defined using T-SQL syntax as follows:

```
EVENTID-bigint,
recordid-bigint,
changetype-int,
eventtime-datetime,
oldvalue-bigint,
newvaluedata-sql_variant,
newvaluechecksum-int,
objecttype-int,
itemsecurity-int,
changedlinkid-bigint,
changedcolleagueid-bigint,
changedmembergroupid-bigint,
changedpropertyid-bigint,
```
**ChangedSourceId**: A unique identifier for the user profile change event.

**RecordId**: An identifier for the user that created the user profile change event.

**ChangeType**: A numeric value representing the user profile Change Type of the user profile event.

**EventTime**: The date and time when the user profile change event occurred.

**OldValue**: This value MUST be NULL.

**NewValueData**: The current value.

**NewValueChecksum**: Numeric value as returned by the T-SQL CHECKSUM function of the actual value.

**ObjectType**: The Object Type of the user profile event.

**ItemSecurity**: The Privacy Policy Type value. This value MUST NOT be NULL if the Object Type is either of the values QuickLink or PersonalizationSite.

**ChangedLinkId**: A unique identifier for the changed link. This value MUST NOT be NULL if the Object Type is either of the values QuickLink or PersonalizationSite.

**ChangedColleagueId**: A record identifier of the Colleague property that has changed. This value MUST NOT be NULL if the Object Type is the value Colleague.

**ChangedMemberGroupId**: A record identifier of a Member Group which corresponds to the membership of the specified user which changed. This value MUST NOT be NULL if the Object Type is either of the values DLMembership or SiteMembership.

**ChangedPropertyId**: The unique identifier for user profile property that has changed. This value MUST NOT be NULL if the Object Type is either of the values SingleValueProperty or MultiValueProperty.

**ChangedSourceId**: A GUID for a Privacy Policy record. This value MUST NOT be NULL for all Object Type values, with the exception of the values WebLog and UserProfile.

### 3.1.5.4 profile_DeleteUserEvents

The **profile_DeleteUserEvents** stored procedure is called to delete all user profile change events older than a given date and time.

**profile_DeleteUserEvents** is defined using T-SQL syntax as follows:

```sql
PROCEDURE profile_DeleteUserEvents (  @MinEventTime                        datetime  );
```

@MinEventTime: A **datetime** value which specifies the date and time prior to which all user profile change events MUST be removed from the database table. This parameter MUST NOT be NULL.

**Return Code Values**: **profile_DeleteUserEvents** MUST return an integer return code that represents the number of user profile change events that were removed from the database table.
Result Sets: profile_DeleteUserEvents MUST NOT return any result set.

3.1.5.5 profile_GetUserColleagueEvents

The profile_GetUserColleagueEvents stored procedure is called to retrieve user profile change events for all Colleague properties of a specified user.

The profile_GetUserColleagueEvents stored procedure is defined using T-SQL syntax as follows:

```
PROCEDURE profile_GetUserColleagueEvents (  
    @RecordId                             bigint,  
    @MinEventTime                         datetime,  
    @ChangeTypeMask                       int,  
    @ObjectTypeMask                       int
 );
```

@RecordId: The record identifier of a user. It MUST return the user profile change events for all Colleague properties of the specified user.

@MinEventTime: Specifies a date and time for which each of the returned user profile change events MUST be more recent. If NULL then profile_GetUserColleagueEvents MUST return all available Colleague property-related user profile change events.

@ChangeTypeMask: Specifies which of the user profile Change Type values of the user profile change events MUST be returned.

@ObjectTypeMask: Specifies the Object Type of the user profile change events that MUST be returned.

Return Code Values: profile_GetUserColleagueEvents MUST return 0.

Result Sets: profile_GetUserColleagueEvents MUST return one Event Log result set.

3.1.5.5.1 Event Log Result Set

The Event Log result set contains the user profile change events of the Colleagues property of the specified user, in order from the earliest to the most recent.

The Event Log result set is defined using T-SQL syntax as follows:

```
EventId                               bigint,  
RecordId                              bigint,  
ChangeType                            int,  
EventTime                             datetime,  
OldValue                              bigint,  
NewValueData                          sql_variant,  
NewValueChecksum                      int,  
ObjectType                            int,  
ItemSecurity                          int,  
ChangedLinkId                         bigint,  
ChangedColleagueId                    bigint,  
ChangedMemberGroupId                  bigint,  
ChangedPropertyId                     bigint,  
ChangedSourceId                       uniqueidentifier;
```
**EventId**: A unique identifier for the user profile change event.

**RecordId**: An identifier for the user that created the user profile change event.

**ChangeType**: A numeric value representing the user profile Change Type of the user profile event.

**EventTime**: Specifies the date and time when the user profile change event occurred.

**OldValue**: MUST be NULL.

**NewValueData**: The current value.

**NewValueChecksum**: A numeric value that is the checksum of the actual value as returned by the T-SQL CHECKSUM function.

**ObjectType**: The Object Type of the user profile event.

**ItemSecurity**: The Privacy Policy Type value. This value MUST NOT be NULL if Object Type is either of the values QuickLink or PersonalizationSite.

**ChangedLinkId**: A unique identifier for the changed link. This value MUST NOT be NULL if the Object Type is either of the values QuickLink or PersonalizationSite.

**ChangedColleagueId**: A record identifier of the Colleague property that has changed. This value MUST NOT be NULL if the Object Type is the value Colleague.

**ChangedMemberGroupId**: A record identifier of a Member Group which corresponds to the specified user membership which changed. This value MUST NOT be NULL if the Object Type is either of the values DLMembership or SiteMembership.

**ChangedPropertyId**: The unique identifier for user profile Property that has changed. This value MUST NOT be NULL if the Object Type is either of the values SingleValueProperty or MultiValueProperty.

**ChangedSourceId**: A GUID for a Privacy Policy record. This value MUST NOT be NULL for all Object Type values, with the exception of WebLog or UserProfile.

### 3.1.5.6  profile_GenerateAnniversaryEvents

The **profile_GenerateAnniversaryEvents** stored procedure is called to create a user profile change event for each upcoming Anniversary property of all date properties, unless the Anniversary Privacy Policy Type is disabled.

In this case, a user profile change event MUST NOT be created. Each user profile change event that is created MUST have a user profile Change Type of "Add" and an Object Type of "Anniversary".

If executed multiple times, **profile_GenerateAnniversaryEvents** MUST NOT create more than one user profile change event for the same anniversary.

**profile_GenerateAnniversaryEvents** is defined using T-SQL syntax as follows:

```tsql
PROCEDURE profile_GenerateAnniversaryEvents (  
   @DaysAheadToScan int  
);
```

**@DaysAheadToScan**: specifies the number of days, starting from the current date, to scan ahead for an anniversary. This parameter MUST be 3.
Return Code Values: `profile GenerateAnniversaryEvents` returns an integer return code which MUST return 0.

Result Sets: `profile GenerateAnniversaryEvents` MUST NOT return any result set.

### 3.1.5.7 `profile GetCurrentChangeToken`

The `profile GetCurrentChangeToken` stored procedure is called to retrieve the user profile Change Token of the most recent user profile change event.

`profile GetCurrentChangeToken` is defined using T-SQL syntax as follows:

```sql
PROCEDURE profile GetCurrentChangeToken();
```

Return Code Values: `profile GetCurrentChangeToken` MUST return 0.

Result Sets: `profile GetCurrentChangeToken` MUST return one result set.

#### 3.1.5.7.1 `profile GetCurrentChangeToken Result Set`

The `profile GetCurrentChangeToken` result set MUST return one record identifying the user profile Change Token of the most recent user profile change event.

The `profile GetCurrentChangeToken` result set is defined using T-SQL syntax as follows:

```sql
(EventTime, EventId)
EventTime datatime,
EventId bigint;
```

- **EventTime**: The date and time of the most recent user profile change event.
- **EventId**: The unique identifier of the most recent user profile change event.

### 3.1.6 Timer Events

None.

### 3.1.7 Other Local Events

None.
4 Protocol Examples

4.1 Sample Data

In the following examples, the user profile store contains five user profiles representing five different users.

<table>
<thead>
<tr>
<th>User</th>
<th>Colleague List</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2, 3</td>
</tr>
<tr>
<td>2</td>
<td>1, 4, 5</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>2, 5</td>
</tr>
<tr>
<td>5</td>
<td>2, 4</td>
</tr>
</tbody>
</table>

The property store contains four properties and their Privacy Policy Type values.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>String</td>
<td>Enabled</td>
</tr>
<tr>
<td>Address</td>
<td>String</td>
<td>Enabled</td>
</tr>
<tr>
<td>Birthday</td>
<td>Date</td>
<td>Enabled</td>
</tr>
<tr>
<td>Marriage Date</td>
<td>Date</td>
<td>Enabled</td>
</tr>
</tbody>
</table>

The user profile property store contains nine user profile property values.

<table>
<thead>
<tr>
<th>User</th>
<th>Property</th>
<th>Value</th>
<th>Privacy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Name</td>
<td>User1</td>
<td>Public</td>
</tr>
<tr>
<td>1</td>
<td>Address</td>
<td>123 New Road, New City, ST</td>
<td>Manager</td>
</tr>
<tr>
<td>1</td>
<td>Marriage Data</td>
<td>02/29/2008</td>
<td>Public</td>
</tr>
<tr>
<td>2</td>
<td>Name</td>
<td>One More User2</td>
<td>Public</td>
</tr>
<tr>
<td>3</td>
<td>Name</td>
<td>Different User3</td>
<td>Public</td>
</tr>
<tr>
<td>3</td>
<td>Birthday</td>
<td>01/02/1973</td>
<td>Contacts</td>
</tr>
<tr>
<td>4</td>
<td>Name</td>
<td>Another User4</td>
<td>Public</td>
</tr>
<tr>
<td>4</td>
<td>Marriage Data</td>
<td>02/03/1974</td>
<td>Private</td>
</tr>
<tr>
<td>5</td>
<td>Name</td>
<td>Last User5</td>
<td>Public</td>
</tr>
<tr>
<td>5</td>
<td>Address</td>
<td>456 Some Road, Some City, ST</td>
<td>Organization</td>
</tr>
</tbody>
</table>

The user profile change entry log contains five user profile change entries.
<table>
<thead>
<tr>
<th>EventId</th>
<th>User</th>
<th>Change Type</th>
<th>ObjectType</th>
<th>Property Name</th>
<th>Value</th>
<th>Change Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Modify</td>
<td>SingleValue Property</td>
<td>Address</td>
<td>123 New Road, New City, ST</td>
<td>02/13/2008 1:23:45 PM</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Add</td>
<td>Colleague</td>
<td>Another User4</td>
<td></td>
<td>02/13/2008 2:34:56 PM</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>Add</td>
<td>Colleague</td>
<td>One More User2</td>
<td></td>
<td>02/13/2008 2:34:56 PM</td>
</tr>
</tbody>
</table>
| 4       | 5    | Add         | WebLog       |               | <?xml version="1.0" encoding="utf-16">  
|         |      |             |              |               | WebLog>  
|         |      |             |              |               |   <Title>  
|         |      |             |              |               |      My New Post  
|         |      |             |              |               | </Title>  
|         |      |             |              |               | <Permalink>  
|         |      |             |              |               | http://site/p5/newpost  
|         |      |             |              |               | </Permalink>  
|         |      |             |              |               | </WebLog>  
|         |      |             |              |               |                                             | 02/13/2008 3:45:07 PM |
| 5       | 3    | Remove      | WebLog       |               | <?xml version="1.0" encoding="utf-16">  
|         |      |             |              |               | WebLog>  
|         |      |             |              |               |   <Title>  
|         |      |             |              |               |      My Old Post  
|         |      |             |              |               | </Title>  
|         |      |             |              |               | <Permalink>  
|         |      |             |              |               | http://site/p3/oldpost  
|         |      |             |              |               | </Permalink>  
|         |      |             |              |               | </WebLog>  
|         |      |             |              |               |                                           | 02/13/2008 4:56:18 PM |
| 6       | 1    | Add         | SingleValue Property | Marriage Data | 02/29/2008 3:21:17 PM |

### 4.2 Retrieve All Changes for the Colleagues of a User

One of the possible scenarios is when it is necessary to retrieve user profile change events of the Colleague properties of a specified user. The following steps can be taken:

- With the user name, use the `membership_getColleagueSuggestions` stored procedure (specified in [MS-UPSPROF], section 3.1.4.3) to get the user identifier (`RecordId`).
- Call `profile_GetUserColleagueEvents (RecordId, Date, ChangeTypeMask, ObjectTypeMask)`

For example, suppose an operation to get all "Add" and "Modify" events for the Colleague properties of "User1" since "02/13/2008 2:00:00 PM". The steps to be taken would be:

- Get "1" from the user profile store using `membership_getColleagueSuggestions [MS-UPSPROF]`.
- Call `profile_GetUserColleagueEvents (1, 02/13/2008 2:00:00 PM, 0x03, 0x03FF).`
In this example, a simplified version of the result set is returned from profile_GetUserColleagueEvents, with only a subset of the columns

The returned result set would consist of:

<table>
<thead>
<tr>
<th>EventId</th>
<th>User</th>
<th>Change Type</th>
<th>ObjectType</th>
<th>Value</th>
<th>Change Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>Add</td>
<td>Colleague</td>
<td>Another User4</td>
<td>02/13/2008 2:34:56 PM</td>
</tr>
</tbody>
</table>

### 4.3 Retrieve All Changes for a Specified User

Using the stored procedures in this protocol, it is possible to retrieve the user profile change events that have occurred for a specific user.

This can be accomplished by calling profile_GetUserEvents. For example:

- Retrieve the RecordId for user 1.
- Retrieve the ViewerRights for user 1.
- Call GetUserEvents with these parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>@RecordId</td>
<td>The RecordId for user 1</td>
</tr>
<tr>
<td>@ViewerRights</td>
<td>The ViewerRights for user 1</td>
</tr>
<tr>
<td>@MinEventId</td>
<td>NULL</td>
</tr>
<tr>
<td>@MinEventTime</td>
<td>02/01/2008 12:00:00 AM</td>
</tr>
<tr>
<td>@ChangeTypeMask</td>
<td>0x07</td>
</tr>
<tr>
<td>@ObjectTypeMask</td>
<td>0x0100</td>
</tr>
</tbody>
</table>

This call would return the following result set. (Non-relevant columns have been omitted for clarity.)

<table>
<thead>
<tr>
<th>RecordId</th>
<th>Change Type</th>
<th>EventTime</th>
<th>NewValueData</th>
<th>ItemSecurity</th>
</tr>
</thead>
<tbody>
<tr>
<td>RecordId for user 1</td>
<td>0x02</td>
<td>02/13/2008 1:23:45 PM</td>
<td>123 New Road, New City, ST</td>
<td>0x08</td>
</tr>
<tr>
<td>RecordId for user 1</td>
<td>0x01</td>
<td>03/01/2008 3:21:17 PM</td>
<td>02/29/2008</td>
<td>0x01</td>
</tr>
</tbody>
</table>

### 4.4 Synchronization

This protocol can be used to maintain the data synchronization between two or more servers. By calling some of these protocol procedures periodically, it is possible for a server to keep track of the latest changes on the other servers. In this example, Server A gets the most recent changes from Server B. The following steps are applied:
- Server A retrieves the date and time of its most recent user profile change event (lastChangeTime).
- Call profile_GetUserEvents (NULL, NULL, NULL, lastChangeTime, 0x0F, 0x03FF).
- If the number of user profile change events returned is less than 1000, then Server A applies those records. Otherwise, Server A synchronizes all of the data with Server B.

Suppose that Server B contains all the data from the example data section, and Server A contains only the first three user profile change events in its user profile change log. The steps to be taken to synchronize Server A and Server B would be as follows:

- **Server A** gets "02/13/2008 2:23:56 PM" from user profile change event with the **EventId** 3.
- **Server A** calls profile_GetUserEvents (NULL, NULL, NULL, "02/13/2008 2:23:56 PM", 0x0F, 0x03FF ) on Server B.

The following example result set would be returned.

In this example, only a subset of the returned columns is being displayed.

<table>
<thead>
<tr>
<th>EventId</th>
<th>User</th>
<th>Change Type</th>
<th>ObjectType</th>
<th>Value</th>
<th>Change Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>5</td>
<td>Add</td>
<td>WebLog</td>
<td>&lt;?xml version=&quot;1.0&quot; encoding=&quot;utf-16&quot;?&gt;&lt;WebLog&gt;&lt;Title&gt;My New Post&lt;/Title&gt;&lt;Permalink&gt;<a href="http://site/p5/newpost">http://site/p5/newpost</a>&lt;/Permalink&gt;&lt;/WebLog&gt;</td>
<td>02/13/2008 3:45:07 PM</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>Remove</td>
<td>WebLog</td>
<td>&lt;?xml version=&quot;1.0&quot; encoding=&quot;utf-16&quot;?&gt;&lt;WebLog&gt;&lt;Title&gt;My Old Post&lt;/Title&gt;&lt;Permalink&gt;<a href="http://site/p3/oldpost">http://site/p3/oldpost</a>&lt;/Permalink&gt;&lt;/WebLog&gt;</td>
<td>02/13/2008 4:56:18 PM</td>
</tr>
</tbody>
</table>
5 Security

5.1 Security Considerations for Implementers

This protocol supports the Security Support Provider Interface (SSPI) and SQL authentication with the protocol server role. These authentication methods are defined in [MS-TDS].

5.2 Index of Security Parameters

None.
6 Appendix A: Product Behavior

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

- Microsoft® Office SharePoint® Server 2007

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
  server 13
Applicability 8
Attribute groups - overview 12
Attributes - overview 12

B

Binary structures - overview 11
Bit fields - overview 11

C

Capability negotiation 8
Change tracking 29
Change type simple type 9
Common data types overview 9
Complex types
  WebLogPostDescription 11
  Complex types - overview 11

D

Data model - abstract
  server 13
Data types
  change type simple type 9
  common 9
  object type simple type 9
  privacy policy type simple type 9
  value simple type 10
Data types - simple
  change type 9
  object type 9
  privacy policy type 9
  value 10

E

Elements - overview 12
Events
  local - server 22
  timer - server 22
Examples
  overview 23
  retrieve all changes for a specified user 25
  retrieve all changes for the colleagues of a user 24
  sample data 23
  synchronization 25

F

Fields - vendor-extensible 8
Flag structures - overview 11

G

Glossary 5
Groups - overview 12

H

Higher-layer triggered events
  server 14

I

Implementer - security considerations 27
Index of security parameters 27
Informative references 6
Initialization
  server 14
  Introduction 5

L

Local events
  server 22

M

Message processing
  server 14
Messages
  attribute groups 12
  attributes 12
  binary structures 11
  bit fields 11
  common data types 9
  complex types 11
  elements 12
  flag structures 11
  groups 12
  simple types 11
  namespaces 11
  result sets 11
  simple types 11
  table structures 11
  transport 9
  view structures 11
  WebLogPostDescription complex type 11
  XML structures 11

Methods
  profile_AddWebLogEvent 14
  profile_DeleteUserEvents 19
  profile_GenerateAnniversaryEvents 21
  profile_GetCurrentChangeToken 22
  profile_GetUserColleagueEvents 20
  profile_GetUserEvents 15
  profile_RemoveWebLogEvent 15

N

Namespaces 11