

# [MS-UPSCSP]: User Profile Service Application Caching Stored Procedures Protocol Specification

---

## Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

## Revision Summary

Date	Revision History	Revision Class	Comments
07/13/2009	0.1	Major	Initial Availability
08/28/2009	0.2	Editorial	Revised and edited the technical content
11/06/2009	0.3	Editorial	Revised and edited the technical content
02/19/2010	1.0	Major	Updated and revised the technical content
03/31/2010	1.01	Editorial	Revised and edited the technical content
04/30/2010	1.02	Editorial	Revised and edited the technical content
06/07/2010	1.03	Editorial	Revised and edited the technical content
06/29/2010	1.04	Editorial	Changed language and formatting in the technical content.
07/23/2010	1.05	Minor	Clarified the meaning of the technical content.
09/27/2010	1.05	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	1.05	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	1.05	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	1.05	No change	No changes to the meaning, language, or formatting of the technical content.
06/10/2011	1.05	No change	No changes to the meaning, language, or formatting of the technical content.

# Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>5</b>
1.1	Glossary .....	5
1.2	References.....	5
1.2.1	Normative References .....	5
1.2.2	Informative References .....	6
1.3	Protocol Overview (Synopsis) .....	6
1.4	Relationship to Other Protocols.....	6
1.5	Prerequisites/Preconditions .....	6
1.6	Applicability Statement.....	6
1.7	Versioning and Capability Negotiation.....	7
1.8	Vendor-Extensible Fields.....	7
1.9	Standards Assignments .....	7
<b>2</b>	<b>Messages.....</b>	<b>8</b>
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
2.2.3	Binary Structures .....	8
2.2.4	Result Sets .....	8
2.2.4.1	profile_GetBulkUserProfileData.ResultSet0 .....	8
2.2.4.2	profile_GetBulkUserProfileData.ResultSet1 .....	9
2.2.4.3	profile_GetBulkUserProfileDataByRecordID.ResultSet0 .....	9
2.2.4.4	profile_GetBulkUserProfileDataByUserID.ResultSet0.....	10
2.2.4.5	profile_GetUserProfileDataBySID.ResultSet0 .....	11
2.2.5	Tables and Views .....	12
2.2.6	XML Structures .....	12
2.2.6.1	Namespaces .....	12
2.2.6.2	Simple Types .....	12
2.2.6.3	Complex Types.....	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
<b>3</b>	<b>Protocol Details.....</b>	<b>13</b>
3.1	Server Details .....	13
3.1.1	Abstract Data Model .....	13
3.1.2	Timers .....	13
3.1.3	Initialization .....	13
3.1.4	Higher-Layer Triggered Events.....	13
3.1.5	Message Processing Events and Sequencing Rules.....	13
3.1.5.1	profile_GetBulkUserProfileDataByEmail.....	13
3.1.5.2	profile_GetBulkUserProfileDataByNTName .....	14
3.1.5.3	profile_GetBulkUserProfileDataByRecordID .....	14
3.1.5.4	profile_GetBulkUserProfileDataByUserID.....	15
3.1.5.5	profile_GetUserProfileDataBySID .....	16
3.1.6	Timer Events .....	16
3.1.7	Other Local Events .....	16
3.2	Client Details.....	16

3.2.1	Abstract Data Model .....	16
3.2.2	Timers .....	16
3.2.3	Initialization .....	16
3.2.4	Higher-Layer Triggered Events .....	17
3.2.5	Message Processing Events and Sequencing Rules .....	17
3.2.6	Timer Events .....	17
3.2.7	Other Local Events .....	17
<b>4</b>	<b>Protocol Examples .....</b>	<b>18</b>
4.1	Retrieving multiple user profiles by their NT Names .....	18
4.2	Retrieving multiple user profiles by their e-mail addresses .....	19
4.3	Retrieving multiple user profiles by their user profile identifiers .....	19
4.4	Retrieving a user profile by its security identifier.....	20
4.5	Retrieving multiple user profiles by their record identifiers .....	20
<b>5</b>	<b>Security .....</b>	<b>21</b>
5.1	Security Considerations for Implementers.....	21
5.2	Index of Security Parameters .....	21
<b>6</b>	<b>Appendix A: Product Behavior .....</b>	<b>22</b>
<b>7</b>	<b>Change Tracking.....</b>	<b>23</b>
<b>8</b>	<b>Index .....</b>	<b>24</b>

# 1 Introduction

This document specifies the User Profile Service Application Caching Stored Procedures Protocol. This protocol enables a protocol client to retrieve the user information stored in a user profile store on a site.

## 1.1 Glossary

The following terms are defined in [\[MS-GLOS\]](#):

**Coordinated Universal Time (UTC)**  
**GUID**  
**security identifier (SID)**

The following terms are defined in [\[MS-OFCGLOS\]](#):

**back-end database server**  
**e-mail address**  
**front-end Web server**  
**master account**  
**profile subtype**  
**request identifier**  
**result set**  
**return code**  
**Security Account Manager (SAM)**  
**Session Initiation Protocol (SIP) address**  
**stored procedure**  
**Transact-Structured Query Language (T-SQL)**  
**Uniform Resource Identifier (URI)**  
**user profile**  
**user profile record identifier**  
**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

### 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](mailto:dochelp@microsoft.com). We will assist you in finding the relevant information. Please check the archive site, <http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624>, as an additional source.

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

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

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

[MS-UPSPROF2] Microsoft Corporation, "[User Profile Stored Procedures Version 2 Protocol Specification](#)".

[MS-WSSFO2] Microsoft Corporation, "[Windows SharePoint Services \(WSS\): File Operations Database Communications Version 2 Protocol Specification](#)".

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

### 1.2.2 Informative References

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

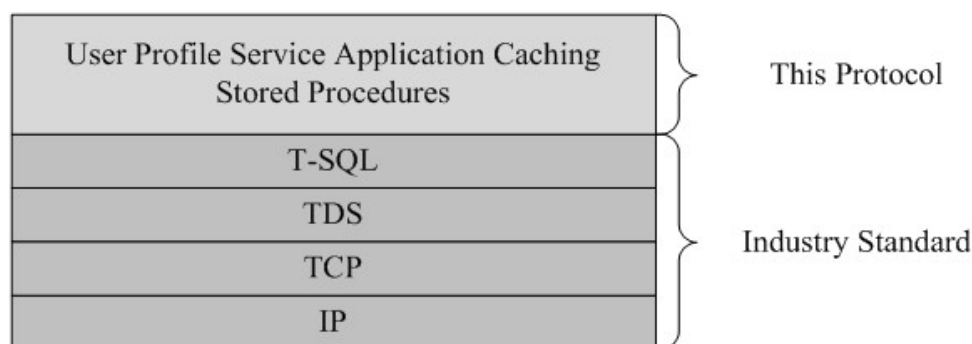
[MS-OFCGLOS] Microsoft Corporation, "[Microsoft Office Master Glossary](#)".

## 1.3 Protocol Overview (Synopsis)

This protocol enables a protocol client to retrieve user information stored in the **user profile store**. It allows the protocol client to pass in a list of **e-mail addresses**, user names, **GUID** identifiers, **user profile record identifiers** or **security identifiers (SIDs)** and returns the **user profile** data.

This protocol can be used in scenarios which require access to user profile data of multiple users in one call.

## 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 for the retrieval of user profile data of multiple users. It can retrieve user profiles of as many users as can be specified via the input parameters and allowed by **T-SQL**.

## 1.7 Versioning and Capability Negotiation

Versions of the data structures or stored procedures contained in the database are expected to be the same as those on 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 [\[MS-WSSFO2\]](#) 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 (Structured Query Language) tables, get **return codes**, and return **result sets**.

### 2.2 Common Data Types

None.

#### 2.2.1 Simple Data Types and Enumerations

None.

#### 2.2.2 Bit Fields and Flag Structures

None.

#### 2.2.3 Binary Structures

None.

#### 2.2.4 Result Sets

##### 2.2.4.1 profile\_GetBulkUserProfileData.ResultSet0

The profile\_GetBulkUserProfileData.ResultSet0 result set returns the basic user profile properties corresponding to the given set of input parameters.

```
LookupValue nvarchar(256),
RecordID bigint,
UserID uniqueidentifier,
NTName nvarchar(400),
PreferredName nvarchar(256),
Email nvarchar(256),
SID varbinary(512),
Manager nvarchar(400),
SipAddress nvarchar(250),
LastUpdate datetime,
MasterRecordID bigint,
ProfileSubtypeID int,
PictureUrl nvarchar(max),
PartitionID uniqueidentifier,
```

**LookupValue:** The e-mail address or the user name used to retrieve the user profile data.

**RecordID:** A record identifier of the user profile. There MUST ONLY be one row corresponding to a given record identifier in this result set.

**UserID:** A GUID identifier of the user profile.

**NTName:** A user name associated with the user profile.



**PreferredName:** The name of the entity as specified in the user profile.

**Email:** An e-mail address associated with the user profile.

**SID:** A security identifier (SID) associated with the user profile.

**Manager:** The user name of the manager of the user as specified in the user profile.

**SipAddress:** Contains the **Session Initiation Protocol (SIP) address** associated with the user profile.

**LastUpdate:** A **UTC** value specifying the last time the user profile was updated.

**MasterRecordID:** Contains the identifier of the **master account** associated with the user profile. If this row is the master account itself, this value **MUST** be the same as RecordID.

**ProfileSubtypeID:** Contains the value identifier of the **profile subtype**.

**PictureUrl:** A **URI** pointing to the picture associated with the user profile.

**PartitionID:** A GUID used to filter the current request. This value **MUST NOT** be null or empty.

#### 2.2.4.2 profile\_GetBulkUserProfileData.ResultSet1

The profile\_GetBulkUserProfileData.ResultSet1 result set returns the user profile properties corresponding to the given set of input parameters.

```
RecordID bigint,  
PropertyID bigint,  
PropertyVal sql_variant,
```

**RecordID:** A user profile record identifier.

**PropertyID:** The identifier of the user profile property.

**PropertyVal:** The value of the user profile property that is specified by PropertyID.

#### 2.2.4.3 profile\_GetBulkUserProfileDataByRecordID.ResultSet0

The profile\_GetBulkUserProfileDataByRecordID.ResultSet1 result set returns the basic user profile properties corresponding to the given set of input parameters.

```
LookupValue bigint,  
RecordID bigint,  
UserID uniqueidentifier,  
NTName nvarchar(400),  
PreferredName nvarchar(256),  
Email nvarchar(256),  
SID varbinary(512),  
Manager nvarchar(400),  
SipAddress nvarchar(250),  
LastUpdate datetime,  
MasterRecordID bigint,  
ProfileSubtypeID int,  
PictureUrl nvarchar(max),  
PartitionID uniqueidentifier,
```

**LookupValue:** The user profile record identifier used to retrieve the user profile data.

**RecordID:** A record identifier of the user profile. There MUST ONLY be one row corresponding to a given record identifier in this result set.

**UserID:** A GUID identifier of the user profile.

**NTName:** A user name associated with the user profile.

**PreferredName:** The name of the entity as specified in the user profile.

**Email:** An e-mail address associated with the user profile.

**SID:** A security identifier (SID) associated with the user profile.

**Manager:** The user name of the manager of the user as specified in the user profile.

**SipAddress:** Contains the Session Initiation Protocol (SIP) address associated with the user profile.

**LastUpdate:** A UTC value specifying the last time the user profile was updated.

**MasterRecordID:** Contains the identifier of the master account associated with the user profile. If this row is the master account itself, this value MUST be the same as RecordID.

**ProfileSubtypeID:** Contains the value identifier of the profile subtype.

**PictureUrl:** A URI pointing to the picture associated with the user profile.

**PartitionID:** A globally unique identifier (GUID) used to filter the current request. This value MUST NOT be null or empty.

#### 2.2.4.4 profile\_GetBulkUserProfileDataByUserID.ResultSet0

The profile\_GetBulkUserProfileDataByUserID.ResultSet1 result set returns the basic user profile properties corresponding to the given set of input parameters.

```
LookupValue uniqueidentifier,  
RecordID bigint,  
UserID uniqueidentifier,  
NTName nvarchar(400),  
PreferredName nvarchar(256),  
Email nvarchar(256),  
SID varbinary(512),  
Manager nvarchar(400),  
SipAddress nvarchar(250),  
LastUpdate datetime,  
MasterRecordID bigint,  
ProfileSubtypeID int,  
PictureUrl nvarchar(2147483647),  
PartitionID uniqueidentifier,
```

**LookupValue:** The GUID identifier of the user profile used to retrieve the user profile data.

**RecordID:** A record identifier of the user profile. There MUST ONLY be one row corresponding to a given record identifier in this result set.

**UserID:** A GUID identifier of the user profile.

**NTName:** A user name associated with the user profile.

**PreferredName:** The name of the entity as specified in the user profile.

**Email:** An e-mail address associated with the user profile.

**SID:** A security identifier (SID) associated with the user profile.

**Manager:** The user name of the manager of the user as specified in the user profile.

**SipAddress:** Contains the Session Initiation Protocol (SIP) address associated with the user profile.

**LastUpdate:** A UTC value specifying the last time the user profile was updated.

**MasterRecordID:** Contains the identifier of the master account associated with the user profile. If this row is the master account itself, this value **MUST** be the same as RecordID.

**ProfileSubtypeID:** Contains the value identifier of the profile subtype.

**PictureUrl:** A URI pointing to the picture associated with the user profile.

**PartitionID:** A globally unique identifier (GUID) used to filter the current request. This value **MUST NOT** be null or empty.

#### 2.2.4.5 profile\_GetUserProfileDataBySID.ResultSet0

The profile\_GetBulkUserProfileDataBySID.ResultSet1 result set returns the basic user profile properties corresponding to the given set of input parameters.

```
LookupValue varbinary(512),
RecordID bigint,
UserID uniqueidentifier,
NTName nvarchar(400),
PreferredName nvarchar(256),
Email nvarchar(256),
SID varbinary(512),
Manager nvarchar(400),
SipAddress nvarchar(250),
LastUpdate datetime,
MasterRecordID bigint,
ProfileSubtypeID int,
PictureUrl nvarchar(max),
PartitionID uniqueidentifier,
```

**LookupValue:** The security identifier (SID) used to retrieve the user profile data.

**RecordID:** A record identifier of the user profile. There **MUST ONLY** be one row corresponding to a given record identifier in this result set.

**UserID:** A GUID identifier of the user profile.

**NTName:** A user name associated with the user profile.

**PreferredName:** The name of the entity as specified in the user profile.

**Email:** An e-mail address associated with the user profile.

**SID:** A security identifier (SID) associated with the user profile.

**Manager:** The user name of the manager of the user as specified in the user profile.

**SipAddress:** Contains the Session Initiation Protocol (SIP) address associated with the user profile.

**LastUpdate:** A UTC value specifying the last time the user profile was updated.

**MasterRecordID:** Contains the identifier of the master account associated with the user profile. If this row is the master account itself, this value **MUST** be the same as RecordID.

**ProfileSubtypeID:** Contains the value identifier of the profile subtype.

**PictureUrl:** A URI pointing to the picture associated with the user profile.

**PartitionID:** A globally unique identifier (GUID) used to filter the current request. This value **MUST** NOT be null or empty.

## 2.2.5 Tables and Views

None.

## 2.2.6 XML Structures

No common XML Structures are defined in this protocol.

### 2.2.6.1 Namespaces

None.

### 2.2.6.2 Simple Types

None.

### 2.2.6.3 Complex Types

None.

### 2.2.6.4 Elements

None.

### 2.2.6.5 Attributes

None.

### 2.2.6.6 Groups

None.

### 2.2.6.7 Attribute Groups

None.

## 3 Protocol Details

### 3.1 Server Details

None.

#### 3.1.1 Abstract Data Model

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

The abstract data model for this protocol is identical to the model defined in [\[MS-UPSPROF2\]](#) section 3.1.1.

#### 3.1.2 Timers

None.

#### 3.1.3 Initialization

None.

#### 3.1.4 Higher-Layer Triggered Events

None.

#### 3.1.5 Message Processing Events and Sequencing Rules

##### 3.1.5.1 profile\_GetBulkUserProfileDataByEmail

The profile\_GetBulkUserProfileDataByEmail stored procedure is called to retrieve the user profiles corresponding to a list of e-mail addresses or Session Initiation Protocol (SIP) addresses.

```
PROCEDURE profile_GetBulkUserProfileDataByEmail (
    @PartitionID uniqueidentifier
    , @KeyValue nvarchar(4000)
    , @ExtendedPropertyIDs nvarchar(4000)
    , @correlationId uniqueidentifier = null
);
```

**@PartitionID:** A globally unique identifier (GUID) used to filter the current request. This value MUST NOT be null or empty.

**@KeyValue:** A list of e-mail or Session Initiation Protocol (SIP) addresses. The values MUST be separated by ';'. If ';' or '^' is part of the e-mail address, they MUST be escaped using '^' character. This parameter MUST NOT be NULL.

**@ExtendedPropertyIDs:** A list of user profile property identifiers. The values MUST be integers and they MUST be separated by ';'. This parameter MUST NOT be NULL.

**@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 [profile\\_GetBulkUserProfileData.ResultSet0](#)

This stored procedure MUST return a [profile\\_GetBulkUserProfileData.ResultSet1](#)

### 3.1.5.2 profile\_GetBulkUserProfileDataByNTName

The profile\_GetBulkUserProfileDataByNTName stored procedure is called to retrieve the user profiles corresponding to a list of **Security Account Manager (SAM)** account names.

```
PROCEDURE profile_GetBulkUserProfileDataByNTName (  
    @PartitionID uniqueidentifier  
    ,@KeyValue nvarchar(4000)  
    ,@ExtendedPropertyIDs nvarchar(4000)  
    ,@correlationId uniqueidentifier = null  
);
```

**@PartitionID:** A globally unique identifier (GUID) used to filter the current request. This value MUST NOT be null or empty.

**@KeyValue:** A list of Security Account Manager (SAM) account names. The values MUST be separated by ';'. If ';' or '^' is part of the e-mail address, they MUST be escaped using '^' character. This parameter MUST NOT be NULL.

**@ExtendedPropertyIDs:** A list of user profile property identifiers. The values MUST be integers and they MUST be separated by ';'. This parameter MUST NOT be NULL.

**@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 [profile\\_GetBulkUserProfileData.ResultSet0](#)

This stored procedure MUST return a [profile\\_GetBulkUserProfileData.ResultSet1](#)

### 3.1.5.3 profile\_GetBulkUserProfileDataByRecordID

The profile\_GetBulkUserProfileDataByRecordID stored procedure is called to retrieve the user profiles corresponding to a list of user profile record identifiers.

```
PROCEDURE profile_GetBulkUserProfileDataByRecordID (  
    @PartitionID uniqueidentifier  
    ,@KeyValue nvarchar(4000)  
    ,@ExtendedPropertyIDs nvarchar(4000)  
    ,@correlationId uniqueidentifier = null  
);
```

**@PartitionID:** A globally unique identifier (GUID) used to filter the current request. This value MUST NOT be null or empty.

**@KeyValue:** A list of user profile record identifiers. The values MUST be separated by ';'. This parameter MUST NOT be NULL.

**@ExtendedPropertyIDs:** A list of user profile property identifiers. The values MUST be integers and they MUST be separated by ';'. This parameter MUST NOT be NULL.

**@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 [profile\\_GetBulkUserProfileDataByRecordID.ResultSet0](#)

This stored procedure MUST return a [profile\\_GetBulkUserProfileData.ResultSet1](#)

### 3.1.5.4 profile\_GetBulkUserProfileDataByUserID

The profile\_GetBulkUserProfileDataByUserID stored procedure is called to retrieve the user profiles corresponding to a list of user profile identifiers.

```
PROCEDURE profile_GetBulkUserProfileDataByUserID (  
    @PartitionID uniqueidentifier  
    ,@KeyValue nvarchar(4000)  
    ,@ExtendedPropertyIDs nvarchar(4000)  
    ,@correlationId uniqueidentifier = null  
);
```

**@PartitionID:** A globally unique identifier (GUID) used to filter the current request. This value MUST NOT be null or empty.

**@KeyValue:** A list of user profile identifiers. The values MUST be separated by ';'. This parameter MUST NOT be NULL.

**@ExtendedPropertyIDs:** A list of user profile property identifiers. The values MUST be integers and they MUST be separated by ';'. This parameter MUST NOT be NULL.

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

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

**Result Sets:**

For the following combination of parameters,

**@PartitionID:** A valid GUID value that is, 0C37852B-34D0-418E-91C6-2AC25AF4BE5B)

**@KeyValue:** A non-null string containing at least one valid user profile identifier (that is, 6C4DA8CD-0095-4E59-B5D1-4454E74C68F5)

This stored procedure MUST return a [profile\\_GetBulkUserProfileDataByUserID.ResultSet0](#)

This stored procedure MUST return a [profile\\_GetBulkUserProfileData.ResultSet1](#)

### 3.1.5.5 profile\_GetUserProfileDataBySID

The profile\_GetUserProfileDataBySID stored procedure is called to retrieve the user profile corresponding to a user profile SID.

```
PROCEDURE profile_GetUserProfileDataBySID (  
    @PartitionID uniqueidentifier  
    ,@KeyValue varbinary(512)  
    ,@ExtendedPropertyIDs nvarchar(4000)  
    ,@correlationId uniqueidentifier = null  
);
```

**@PartitionID:** A globally unique identifier (GUID) used to filter the current request. This value MUST NOT be null or empty.

**@KeyValue:** A user profile SID. This parameter MUST NOT be NULL.

**@ExtendedPropertyIDs:** A list of user profile property identifiers. The values MUST be integers and they MUST be separated by ';'. This parameter MUST NOT be NULL.

**@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 [profile\\_GetUserProfileDataBySID.ResultSet0](#)

This stored procedure MUST return a [profile\\_GetBulkUserProfileData.ResultSet1](#)

### 3.1.6 Timer Events

None.

### 3.1.7 Other Local Events

None.

## 3.2 Client Details

None.

### 3.2.1 Abstract Data Model

None.

### 3.2.2 Timers

None.

### 3.2.3 Initialization

None.



### **3.2.4 Higher-Layer Triggered Events**

None.

### **3.2.5 Message Processing Events and Sequencing Rules**

None.

### **3.2.6 Timer Events**

None.

### **3.2.7 Other Local Events**

None.

## 4 Protocol Examples

This section provides specific example scenarios for bulk retrieving user profile data using either of the following - NT Names, e-mail addresses, user profile identifiers, security identifier, user profile record identifiers.

### Example Data

In these examples, the data is originally arranged this way:

- The user profile store contains five user profiles, with user names, NTNames, e-mail addresses and user profile identifiers as shown in the following table

User Name	NTName	E-mail address	User profile identifier
Syed Abbas	contoso\syed	syed@contoso.com	4bccfd46-fb01-4c9b-988c-0730eae529a
Lori Kane	contoso\lori	lori@contoso.com	b8110f57-8f8e-4c4a-9570-1c75828e18ef
Tai Yee	contoso\tai	tai@contoso.com	5737a7f5-eb81-4b1f-adae-873e1e762e74
Ellen Adams	contoso\ellen	ellen@contoso.com	7c21b982-643c-4cb4-89c7-09cbe41f030c
Sara Davis	contoso\sara	sara@contoso.com	0a99119e-37fa-445a-b9ba-a6094e0ab52d

The following identifiers are used in the examples in this section:

- PartitionID - 0C37852B-34D0-418E-91C6-2AC25AF4BE5B
- CorrelationID - 54bfa4f5-b4c0-48b8-abf2-601b7cf14117
- Extended Property identifiers as shown in the following table

Extended Property Identifier	Extended Property Name
13	Title
14	Department

- User profile Security Identifier (SID) - 0x010500000000000515000000A065CF7E784B9B5FE77C8770AE0C1F00
- User profile Record Identifiers - 2106, 2107, 2108, 2109 and 2110

### 4.1 Retrieving multiple user profiles by their NT Names

To retrieve the five user profiles by their NT Names, consider the following T-SQL syntax used by the protocol client to call the **profile\_GetBulkUserProfileDataByNTName**:

```
declare @MyPartitionID uniqueidentifier
declare @MyCorrelationID uniqueidentifier
```

```

declare @MyExtendedPropertyIDs nvarchar(4000)
declare @NTNameList nvarchar(4000)
set @MyPartitionID=N'0C37852B-34D0-418E-91C6-2AC25AF4BE5B'
set @MyCorrelationID=N'54bfa4f5-b4c0-48b8-abf2-601b7cf14117'
set @MyExtendedPropertyIDs = N'13;14'
set @NTNameList=N'contoso\syed;contoso\lori;contoso\tai;contoso\ellen;contoso\sara'
exec profile_GetBulkUserProfileDataByNTName
@PartitionID=@MyPartitionID,
@KeyValue=@NTNameList,
@ExtendedPropertyIDs=@MyExtendedPropertyIDs,
@correlationId=@MyCorrelationID

```

## 4.2 Retrieving multiple user profiles by their e-mail addresses

To retrieve the five user profiles by their e-mail addresses, consider the following T-SQL syntax used by the protocol client to call the **profile\_GetBulkUserProfileDataByEmail**:

```

declare @MyPartitionID uniqueidentifier
declare @MyCorrelationID uniqueidentifier
declare @MyExtendedPropertyIDs nvarchar(4000)
declare @EmailList nvarchar(4000)
set @MyPartitionID=N'0C37852B-34D0-418E-91C6-2AC25AF4BE5B'
set @MyCorrelationID=N'54bfa4f5-b4c0-48b8-abf2-601b7cf14117'
set @MyExtendedPropertyIDs = N'13;14'
set @EmailList = N'syed@contoso.com;lori@contoso.com;tai@contoso.com;ellen@contoso.com;sara@contoso.com'
exec profile_GetBulkUserProfileDataByEmail
@PartitionID=@MyPartitionID,
@KeyValue=@EmailList,
@ExtendedPropertyIDs=@MyExtendedPropertyIDs,
@correlationId=@MyCorrelationID

```

## 4.3 Retrieving multiple user profiles by their user profile identifiers

To retrieve the five user profiles by their user profile identifiers, consider the following T-SQL syntax used by the protocol client to call the **proc\_GetBulkUserProfileDataByUserID**:

```

declare @MyPartitionID uniqueidentifier
declare @MyCorrelationID uniqueidentifier
declare @MyExtendedPropertyIDs nvarchar(4000)
declare @UserIDList nvarchar(4000)
set @MyPartitionID=N'0C37852B-34D0-418E-91C6-2AC25AF4BE5B'
set @MyCorrelationID=N'54bfa4f5-b4c0-48b8-abf2-601b7cf14117'
set @MyExtendedPropertyIDs = N'13;14'
set @UserIDList=N'4bccfd46-fb01-4c9b-988c-0730eae529a;b8110f57-8f8e-4c4a-9570-1c75828e18ef;5737a7f5-eb81-4b1f-adae-873e1e762e74;7c21b982-643c-4cb4-89c7-09cbe41f030c;0a99119e-37fa-445a-b9ba-a6094e0ab52d'
exec profile_GetBulkUserProfileDataByUserID
@PartitionID=@MyPartitionID,
@KeyValue=@UserIDList,
@ExtendedPropertyIDs=@MyExtendedPropertyIDs,
@correlationId=@MyCorrelationID

```

## 4.4 Retrieving a user profile by its security identifier

To retrieve a user profile by its security identifier, consider the following T-SQL syntax used by the protocol client to call the **profile\_GetUserProfileDataBySID**:

```
declare @MyPartitionID uniqueidentifier
declare @MyCorrelationID uniqueidentifier
declare @MyExtendedPropertyIDs nvarchar(4000)
declare @SID varbinary(512)
set @MyPartitionID=N'0C37852B-34D0-418E-91C6-2AC25AF4BE5B'
set @MyCorrelationID=N'54bfa4f5-b4c0-48b8-abf2-601b7cf14117'
set @MyExtendedPropertyIDs = N'13;14'
set @SID=0x010500000000000515000000A065CF7E784B9B5FE77C8770AE0C1F00
exec profile_GetUserProfileDataBySID
@PartitionID=@MyPartitionID,
@KeyValue=@SID,
@ExtendedPropertyIDs=@MyExtendedPropertyIDs,
@correlationId=@MyCorrelationID
```

## 4.5 Retrieving multiple user profiles by their record identifiers

To retrieve the five user profiles by their user profile record identifiers, consider the following T-SQL syntax used by the protocol client to call the **profile\_GetBulkUserProfileDataByRecordID**:

```
declare @MyPartitionID uniqueidentifier
declare @MyCorrelationID uniqueidentifier
declare @MyExtendedPropertyIDs nvarchar(4000)
declare @RecordIDList nvarchar(4000)
set @MyPartitionID=N'0C37852B-34D0-418E-91C6-2AC25AF4BE5B'
set @MyCorrelationID=N'54bfa4f5-b4c0-48b8-abf2-601b7cf14117'
set @MyExtendedPropertyIDs = N'13;14'
set @RecordIDList = N'2110;2109;2108;2107;2106'
exec profile_GetBulkUserProfileDataByRecordID
@PartitionID=@MyPartitionID,
@KeyValue=@RecordIDList,
@ExtendedPropertyIDs=@MyExtendedPropertyIDs,
@correlationId=@MyCorrelationID
```

## **5 Security**

### **5.1 Security Considerations for Implementers**

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

### **5.2 Index of Security Parameters**

None.

## 6 Appendix A: Product Behavior

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

- Microsoft® SharePoint® Server 2010

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

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

## 7 Change Tracking

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

## 8 Index

### A

Abstract data model

[client](#) 16

[server](#) 13

[Applicability](#) 6

[Attribute groups - overview](#) 12

[Attributes - overview](#) 12

### B

[Binary structures - overview](#) 8

[Bit fields - overview](#) 8

### C

[Capability negotiation](#) 7

[Change tracking](#) 23

Client

[abstract data model](#) 16

[higher-layer triggered events](#) 17

[initialization](#) 16

[local events](#) 17

[message processing](#) 17

[sequencing rules](#) 17

[timer events](#) 17

[timers](#) 16

Common data types

[overview](#) 8

[Complex types - overview](#) 12

### D

Data model - abstract

[client](#) 16

[server](#) 13

Data types

[common](#) 8

Data types - simple

[overview](#) 8

### E

[Elements - overview](#) 12

Events

[local - client](#) 17

[local - server](#) 16

[timer - client](#) 17

[timer - server](#) 16

Examples

[overview](#) 18

[retrieving multiple user profiles by their e-mail](#)

[addresses](#) 19

[retrieving multiple user profiles by their NT](#)

[Names](#) 18

[retrieving multiple user profiles by their record](#)

[identifiers](#) 20

[retrieving multiple user profiles by their security](#)

[identifier](#) 20

[retrieving multiple user profiles by their user](#)  
[profile identifiers](#) 19

### F

[Fields - vendor-extensible](#) 7

[Flag structures - overview](#) 8

### G

[Glossary](#) 5

[Groups - overview](#) 12

### H

Higher-layer triggered events

[client](#) 17

[server](#) 13

### I

[Implementer - security considerations](#) 21

[Index of security parameters](#) 21

[Informative references](#) 6

Initialization

[client](#) 16

[server](#) 13

[Introduction](#) 5

### L

Local events

[client](#) 17

[server](#) 16

### M

Message processing

[client](#) 17

[server](#) 13

Messages

[attribute groups](#) 12

[attributes](#) 12

[binary structures](#) 8

[bit fields](#) 8

[common data types](#) 8

[complex types](#) 12

[elements](#) 12

[enumerations](#) 8

[flag structures](#) 8

[groups](#) 12

[namespaces](#) 12

[profile\\_GetBulkUserProfileData.ResultSet0 result](#)  
[set](#) 8

[profile\\_GetBulkUserProfileData.ResultSet1 result](#)  
[set](#) ([section 2.2.4.2](#) 9, [section 2.2.4.3](#) 9,  
[section 2.2.4.4](#) 10, [section 2.2.4.5](#) 11)

[simple data types](#) 8

[simple types](#) 12



- [table structures](#) 12
- [transport](#) 8
- [view structures](#) 12
- [XML structures](#) 12
- Methods
  - [profile\\_GetBulkUserProfileDataByEmail](#) 13
  - [profile\\_GetBulkUserProfileDataByNTName](#) 14
  - [profile\\_GetBulkUserProfileDataByRecordID](#) 14
  - [profile\\_GetBulkUserProfileDataByUserID](#) 15
  - [profile\\_GetUserProfileDataBySID](#) 16
- N**
  - [Namespaces](#) 12
  - [Normative references](#) 5
- O**
  - [Overview \(synopsis\)](#) 6
- P**
  - [Parameters - security index](#) 21
  - [Preconditions](#) 6
  - [Prerequisites](#) 6
  - [Product behavior](#) 22
  - [profile\\_GetBulkUserProfileData.ResultSet0 result set](#) 8
  - [profile\\_GetBulkUserProfileData.ResultSet1 result set](#) (section 2.2.4.2 9, section 2.2.4.3 9, section 2.2.4.4 10, section 2.2.4.5 11)
  - [profile\\_GetBulkUserProfileDataByEmail method](#) 13
  - [profile\\_GetBulkUserProfileDataByNTName method](#) 14
  - [profile\\_GetBulkUserProfileDataByRecordID method](#) 14
  - [profile\\_GetBulkUserProfileDataByUserID method](#) 15
  - [profile\\_GetUserProfileDataBySID method](#) 16
- R**
  - References
    - [informative](#) 6
    - [normative](#) 5
  - [Relationship to other protocols](#) 6
  - Result sets - messages
    - [profile\\_GetBulkUserProfileData.ResultSet0](#) 8
    - [profile\\_GetBulkUserProfileData.ResultSet1](#) (section 2.2.4.2 9, section 2.2.4.3 9, section 2.2.4.4 10, section 2.2.4.5 11)
  - [Retrieving multiple user profiles by their e-mail addresses example](#) 19
  - [Retrieving multiple user profiles by their NT Names example](#) 18
  - [Retrieving multiple user profiles by their record identifiers example](#) 20
  - [Retrieving multiple user profiles by their security identifier example](#) 20
  - [Retrieving multiple user profiles by their user profile identifiers example](#) 19
- S**

- Security
  - [implementer considerations](#) 21
  - [parameter index](#) 21
- Sequencing rules
  - [client](#) 17
  - [server](#) 13
- Server
  - [abstract data model](#) 13
  - [higher-layer triggered events](#) 13
  - [initialization](#) 13
  - [local events](#) 16
  - [message processing](#) 13
  - [profile\\_GetBulkUserProfileDataByEmail method](#) 13
  - [profile\\_GetBulkUserProfileDataByNTName method](#) 14
  - [profile\\_GetBulkUserProfileDataByRecordID method](#) 14
  - [profile\\_GetBulkUserProfileDataByUserID method](#) 15
  - [profile\\_GetUserProfileDataBySID method](#) 16
  - [sequencing rules](#) 13
  - [timer events](#) 16
  - [timers](#) 13
- Simple data types
  - [overview](#) 8
- [Simple types - overview](#) 12
- [Standards assignments](#) 7
- Structures
  - [binary](#) 8
  - [table and view](#) 12
  - [XML](#) 12
- T**
  - [Table structures - overview](#) 12
  - Timer events
    - [client](#) 17
    - [server](#) 16
  - Timers
    - [client](#) 16
    - [server](#) 13
  - [Tracking changes](#) 23
  - [Transport](#) 8
  - Triggered events - higher-layer
    - [client](#) 17
    - [server](#) 13
  - Types
    - [complex](#) 12
    - [simple](#) 12
- V**
  - [Vendor-extensible fields](#) 7
  - [Versioning](#) 7
  - [View structures - overview](#) 12
- X**
  - [XML structures](#) 12