[MS-UPSCWS]:

User Profile Service Application Caching Web Service Protocol

---

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **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 can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.
- **Patents**. Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft Open Specifications Promise or the Microsoft Community Promise. If you would prefer a written license, or if the technologies described in this documentation 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 might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names**. The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are 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 as specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications documentation does 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 documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

Revision Summary

Date	Revision History	Revision Class	Comments
7/13/2009	0.1	Major	Initial Availability
8/28/2009	0.2	Editorial	Revised and edited the technical content
11/6/2009	0.3	Editorial	Revised and edited the technical content
2/19/2010	1.0	Major	Updated and revised the technical content
3/31/2010	1.01	Editorial	Revised and edited the technical content
4/30/2010	1.02	Editorial	Revised and edited the technical content
6/7/2010	1.03	Editorial	Revised and edited the technical content
6/29/2010	1.04	Editorial	Changed language and formatting in the technical content.
7/23/2010	1.05	Minor	Clarified the meaning of the technical content.
9/27/2010	1.05	None	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	1.05	None	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	1.05	None	No changes to the meaning, language, or formatting of the technical content.
3/18/2011	1.05	None	No changes to the meaning, language, or formatting of the technical content.
6/10/2011	1.05	None	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	2.0	Major	Significantly changed the technical content.
4/11/2012	2.0	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	2.0	None	No changes to the meaning, language, or formatting of the technical content.
9/12/2012	2.0	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	2.1	Minor	Clarified the meaning of the technical content.
2/11/2013	2.1	None	No changes to the meaning, language, or formatting of the technical content.
7/30/2013	3.0	Major	Significantly changed the technical content.
11/18/2013	3.0	None	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	3.1	Minor	Clarified the meaning of the technical content.
4/30/2014	3.2	Minor	Clarified the meaning of the technical content.
7/31/2014	3.2	None	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
10/30/2014	3.2	None	No changes to the meaning, language, or formatting of the technical content.
2/26/2016	4.0	Major	Significantly changed the technical content.
7/15/2016	4.0	None	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction

The User Profile Service Application Caching Web Service Protocol enables a protocol client to retrieve user information from a database that stores user profile data for a site.

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

application server: A computer that provides infrastructure and services for applications that are hosted on a server farm.

authentication: The ability of one entity to determine the identity of another entity.

back-end database server: A server that hosts data, configuration settings, and stored procedures that are associated with one or more applications.

base64 encoding: A binary-to-text encoding scheme whereby an arbitrary sequence of bytes is converted to a sequence of printable ASCII characters, as described in [RFC4648].

colleague: A user who has a social networking relationship with another user.

email address: A string that identifies a user and enables the user to receive Internet messages.

endpoint: A communication port that is exposed by an **application server** for a specific shared service and to which messages can be addressed.

globally unique identifier (GUID): A term used interchangeably with universally unique identifier (UUID) in Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the value. Specifically, the use of this term does not imply or require that the algorithms described in [RFC4122] or [C706] must be used for generating the **GUID**. See also universally unique identifier (UUID).

login name: A string that is used to identify a user or entity to an operating system, directory service, or distributed system. For example, in Windows-integrated authentication, a login name uses the form "DOMAIN\username".

request identifier: A **GUID** that is used to identify a specific action or procedure that is sent to a protocol server or a protocol client.

security identifier (SID): An identifier for security principals in Windows that is used to identify an account or a group. Conceptually, the **SID** is composed of an account authority portion (typically a domain) and a smaller integer representing an identity relative to the account authority, termed the relative identifier (RID). The **SID** format is specified in [MS-DTYP] section 2.4.2; a string representation of **SIDs** is specified in [MS-DTYP] section 2.4.2 and [MS-AZOD] section 1.1.1.2.

Session Initiation Protocol (SIP) address: A URI that does not include a "sip:" prefix and is used to establish multimedia communications sessions between two or more users over an IP network, as described in [RFC3261].

SOAP action: The HTTP request header field used to indicate the intent of the SOAP request, using a **URI** value. See [SOAP1.1] section 6.1.1 for more information.

SOAP body: A container for the payload data being delivered by a SOAP message to its recipient. See [SOAP1.2-1/2007] section 5.3 for more information.

SOAP fault detail: A string containing a human-readable explanation of a SOAP fault, which is not intended for algorithmic processing. See [SOAP1.2-1/2007] section 5.4.5 for more information.

stream: An element of a compound file, as described in [MS-CFB]. A stream contains a sequence of bytes that can be read from or written to by an application, and they can exist only in storages.

Unicode: A character encoding standard developed by the Unicode Consortium that represents almost all of the written languages of the world. The **Unicode** standard [UNICODE5.0.0/2007] provides three forms (UTF-8, UTF-16, and UTF-32) and seven schemes (UTF-8, UTF-16, UTF-16 BE, UTF-16 LE, UTF-32, UTF-32 LE, and UTF-32 BE).

Uniform Resource Identifier (URI): A string that identifies a resource. The URI is an addressing mechanism defined in Internet Engineering Task Force (IETF) Uniform Resource Identifier (URI): Generic Syntax [RFC3986].

user profile: A collection of properties that pertain to a specific person or entity within a portal site.

user profile record identifier: An integer that uniquely identifies a user profile record.

User Profile Service: A data source that stores, provides, and applies information about users.

user profile store: A database that stores information about each user profile.

Web Services Description Language (WSDL): An XML format for describing network services as a set of endpoints that operate on messages that contain either document-oriented or procedure-oriented information. The operations and messages are described abstractly and are bound to a concrete network protocol and message format in order to define an endpoint. Related concrete endpoints are combined into abstract endpoints, which describe a network service. WSDL is extensible, which allows the description of endpoints and their messages regardless of the message formats or network protocols that are used.

WSDL message: An abstract, typed definition of the data that is communicated during a **WSDL operation** [WSDL]. Also, an element that describes the data being exchanged between web service providers and clients.

WSDL operation: A single action or function of a web service. The execution of a WSDL operation typically requires the exchange of messages between the service requestor and the service provider.

XML namespace: A collection of names that is used to identify elements, types, and attributes in XML documents identified in a URI reference [RFC3986]. A combination of XML namespace and local name allows XML documents to use elements, types, and attributes that have the same names but come from different sources. For more information, see [XMLNS-2ED].

XML namespace prefix: An abbreviated form of an **XML namespace**, as described in [XML].

XML schema: A description of a type of XML document that is typically expressed in terms of constraints on the structure and content of documents of that type, in addition to the basic syntax constraints that are imposed by XML itself. An XML schema provides a view of a document type at a relatively high level of abstraction.

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

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

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.

[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

[SOAP1.1] Box, D., Ehnebuske, D., Kakivaya, G., et al., "Simple Object Access Protocol (SOAP) 1.1", May 2000, http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

[SOAP1.2/1] Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J., and Nielsen, H.F., "SOAP Version 1.2 Part 1: Messaging Framework", W3C Recommendation, June 2003, http://www.w3.org/TR/2003/REC-soap12-part1-20030624

[WSDL] Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S., "Web Services Description Language (WSDL) 1.1", W3C Note, March 2001, http://www.w3.org/TR/2001/NOTE-wsdl-20010315

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

[XMLSCHEMA1] Thompson, H., Beech, D., Maloney, M., and Mendelsohn, N., Eds., "XML Schema Part 1: Structures", W3C Recommendation, May 2001, http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/

[XMLSCHEMA2] Biron, P.V., Ed. and Malhotra, A., Ed., "XML Schema Part 2: Datatypes", W3C Recommendation, May 2001, http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/

[XMLSCHEMA] World Wide Web Consortium, "XML Schema", September 2005, http://www.w3.org/2001/XMLSchema

1.2.2 Informative References

[MS-SPSTWS] Microsoft Corporation, "SharePoint Security Token Service Web Service Protocol".

[MS-SPTWS] Microsoft Corporation, "Service Platform Topology Web Service Protocol".

[RFC2822] Resnick, P., Ed., "Internet Message Format", RFC 2822, April 2001, http://www.ietf.org/rfc/rfc2822.txt

1.3 Overview

This protocol allows protocol clients to gain access to **user profile** information through a middle-tier **application server**, instead of a **back-end database server**. A typical scenario is a protocol client fetching user profile properties, such as the name, **email address**, picture URL, or **colleagues** of one or more users, by using user credentials, such as an user name or **security identifier (SID)**, or another field, such as the **user profile record identifier**, that identify those users. The middle-tier application server caches user profile fields that are commonly used and exposes them to protocol clients through this protocol. Consequently, this protocol helps distribute the load from the database tier to the application tier; it helps the protocol server scale to handle higher loads because read

requests for commonly used fields are the operations performed most frequently. In addition, this protocol is designed for bulk operations. Protocol clients can use this protocol to gain access to user profile information for many users simultaneously, and the corresponding response can contain data for as many users as requested. This feature optimizes the efficiency of operations for a large user base.

1.4 Relationship to Other Protocols

This protocol uses the SOAP message protocol for formatting requests and responses as described in [SOAP1.1], [SOAP1.2/1] and [SOAP1.2/2]. It transmits these messages by using HTTP protocol as described in [RFC2616], or the HTTPS protocol as described in [RFC2818] or the TCP protocol as described in [MC-NMF].

The User Profile Service Application Caching Web Service Protocol uses SOAP over HTTP(S) as shown in the following layering diagram.

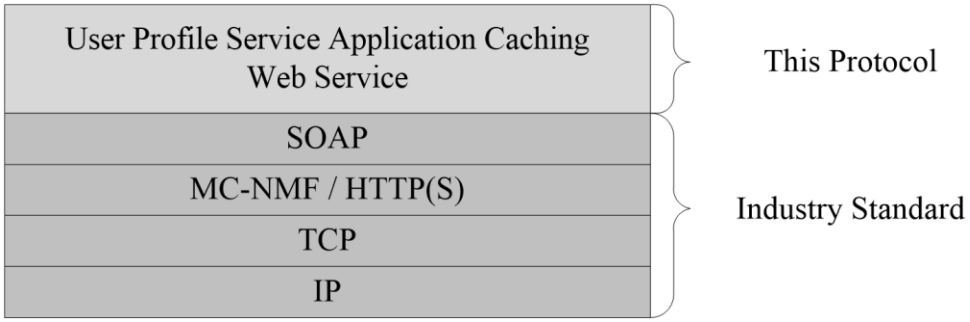

Figure 1: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

This protocol operates against a protocol server that exposes one or more **endpoint URIs** that are known by protocol clients. Both the protocol server endpoint URI and transport are either known by the protocol client or are obtained by using the discovery mechanism described in [MS-SPTWS].

The protocol client can obtain the **ApplicationClassId** and **ApplicationVersion** of the protocol server with which it communicates. The endpoint URI of a protocol server is required to provide the discovery mechanism described in [MS-SPTWS].

This protocol requires the protocol client to have the necessary permission settings to call methods on the protocol server. The protocol also requires that the protocol client implements the token-based security needed for the protocol server and the security protocols described in [MS-SPSTWS].

1.6 Applicability Statement

This protocol is designed to retrieve user profile data for multiple users in one call. The number of users for which that data is retrieved is limited only by the configuration of the transport layer.

1.7 Versioning and Capability Negotiation

Supported Transports: This protocol can be implemented by using transports that support sending SOAP messages, as described in section 2.1.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

In the following sections, the schema definition might differ from the processing rules imposed by the protocol. The **WSDL** in this specification matches the WSDL that shipped with the product and provides a base description of the schema. The text that introduces the WSDL might specify differences that reflect actual Microsoft product behavior. For example, the schema definition might allow for an element to be **empty**, **null**, or **not present** but the behavior of the protocol as specified restricts the same elements to being **non-empty**, **not null**, and **present**.

2.1 Transport

Protocol servers MUST support Simple Object Access Protocol (SOAP) over Hypertext Transfer Protocol (HTTP), Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS), or TCP.

All protocol messages MUST be transported by using HTTP or TCP bindings at the transport level.

Protocol messages MUST be formatted as specified in either [SOAP1.1] section 4 or [SOAP1.2/1] section 5. Protocol server faults MUST be returned by using HTTP status codes, as specified in [RFC2616] section 10, or SOAP faults, as specified in [SOAP1.1] section 4.4 or [SOAP1.2/1] section 5.4.

If the HTTPS transport is used, a server certificate MUST be deployed.

This protocol MAY transmit an additional SOAP header, the **ServiceContext** header, as specified in [MS-SPSTWS].

This protocol does not define any means for activating a protocol server or protocol client. The protocol server MUST be configured and begin listening in an implementation-specific way. In addition, the protocol client MUST know the format and transport that is used by the server, for example, the SOAP format over an HTTP transport.

2.2 Common Message Syntax

This section contains common definitions that are used by this protocol. The syntax of the definitions uses **XML schema**, as specified in [XMLSCHEMA1] and [XMLSCHEMA2], and WSDL, as specified in [WSDL].

2.2.1 Namespaces

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

Prefix	Namespace URI	Reference
q1	http://Microsoft/Office/Server/UserProfiles	
q2	http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.UserProfiles.Cache	
q3	http://schemas.microsoft.com/2003/10/Serialization/Arrays	
q4	http://schemas.datacontract.org/2004/07/System.IO	
q5	http://schemas.datacontract.org/2004/07/System	
soap	http://schemas.xmlsoap.org/wsdl/soap/	[SOAP1.1]

Prefix	Namespace URI	Reference
tns	http://tempuri.org/	
tns1	http://schemas.microsoft.com/2003/10/Serialization/	
tns2	http://tempuri.org/Imports	
wsam	http://www.w3.org/2007/05/addressing/metadata	
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1] [XMLSCHEMA2]

2.2.2 Messages

This specification does not define any common **WSDL message** definitions.

2.2.3 Elements

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

2.2.4 Complex Types

The following table summarizes the set of common XML schema complex type definitions defined by this specification. XML schema complex type definitions that are specific to a particular operation are described with the operation.

Complex type	Description
ArrayOfUserData	ArrayOfUserData is a sequence of zero or more UserData complex types.
ConfigurationFault	The **ConfigurationFault** complex type describes errors that occur because of configuration failures either at the service endpoint of this protocol or in transport layers. It is an extension of the **UserProfileFault** complex type.
InputFault	The **InputFault** complex type describes errors that occur because of invalid **UserSearchCriteria** input from a protocol client. The **InputFault** complex type is an extension of the **UserProfileFault** complex type.
UnknownFault	The **UnknownFault** complex type describes errors that occur for unknown reasons in the service endpoint of this protocol. The **UnknownFault** complex type is an extension of the **UserProfileFault** complex type.
UserData	**UserData** is a complex type that contains the commonly used properties of the user profile data for a specific user in the **user profile store**.
UserProfileFault	The **UserProfileFault** complex type describes errors that occur because of errors within the service endpoint of this protocol.

2.2.4.1 ConfigurationFault

Namespace: http://Microsoft/Office/Server/UserProfiles

The **ConfigurationFault** complex type describes errors that occur because of configuration failures either at the service endpoint of this protocol or in transport layers. It is an extension of the **UserProfileFault** complex type.

```
<xs:complexType name="ConfigurationFault" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexContent mixed="false">
    <xs:extension base="q1:UserProfileFault">
      <xs:sequence/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

2.2.4.2 InputFault

Namespace: http://Microsoft/Office/Server/UserProfiles

The **InputFault** complex type describes errors that occur because of invalid **UserSearchCriteria** input from a protocol client. The **InputFault** complex type is an extension of the **UserProfileFault** complex type.

```
<xs:complexType name="InputFault" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexContent mixed="false">
    <xs:extension base="q1:UserProfileFault">
      <xs:sequence/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

2.2.4.3 UnknownFault

Namespace: http://Microsoft/Office/Server/UserProfiles

The **UnknownFault** complex type describes errors that occur for unknown reasons in the service endpoint of this protocol. The **UnknownFault** complex type is an extension of the **UserProfileFault** complex type.

```
<xs:complexType name="UnknownFault" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexContent mixed="false">
    <xs:extension base="q1:UserProfileFault">
      <xs:sequence/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

2.2.4.4 UserProfileFault

Namespace: http://Microsoft/Office/Server/UserProfiles

The **UserProfileFault** complex type describes errors that occur because of errors within the service endpoint of this protocol.

```
<xs:complexType name="UserProfileFault" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" name="FailureDetail" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="FaultCode" type="q1:FaultCodes"/>
  </xs:sequence>
</xs:complexType>
```

FailureDetail: Specifies an error string. The protocol does not impose any restriction on the behavior of the protocol client with respect to this field.

FaultCode: Specifies a numerical indicator of the error that occurred. The protocol does not impose any restriction on the behavior of the protocol client with respect to this field.

2.2.4.5 ArrayOfUserData

Namespace: http://Microsoft/Office/Server/UserProfiles

ArrayOfUserData is a sequence of zero or more UserData complex types.

```
<xs:complexType name="ArrayOfUserData" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="UserData" nillable="true"
type="q1:UserData"/>
  </xs:sequence>
</xs:complexType>
```

UserData: Specifies a **UserData** complex type.

2.2.4.6 UserData

Namespace: http://Microsoft/Office/Server/UserProfiles

UserData is a complex type that contains the commonly used properties of the user profile data for a specific user in the user profile store.

```
<xs:complexType name="UserData" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" name="Department" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="Email" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="LastUpdate" type="xs:dateTime"/>
    <xs:element minOccurs="0" name="MasterRecordID" type="xs:long"/>
    <xs:element minOccurs="0" name="NTName" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="PartitionID" type="tns1:guid"/>
    <xs:element minOccurs="0" name="PictureUrl" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="PreferredName" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="ProfileSubtypeID" type="xs:int"/>
    <xs:element minOccurs="0" name="RecordID" type="xs:long"/>
    <xs:element minOccurs="0" name="SID" nillable="true" type="xs:base64Binary"/>
    <xs:element minOccurs="0" name="SipAddress" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="Title" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="UserID" type="tns1:guid"/>
    <xs:element minOccurs="0" name="PersonalSpace" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="FeedIdentifier" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="FeedPrivacyActivity" nillable="true" type="xs:int"/>
    <xs:element minOccurs="0" name="IsPeopleListPublic" nillable="true" type="xs:boolean"/>
    <xs:element minOccurs="0" name="EmailOptin" nillable="true" type="xs:int"/>
    <xs:element minOccurs="0" name="StatusNote" nillable="true" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

Department: The department at work to which the user belongs.

Email: The work-related email address of the user.

LastUpdate: The timestamp, in **UTC** format, when the user's user profile data was last modified.

MasterRecordID: The identifier of the primary record that stores the user profile data. If the **MasterRecordID** equals the **RecordID**, then this record MUST be the primary record of the user

profile data of the user. The primary record is the main record for the user if the user has two user profiles.

NTName: The user name string that uniquely identifies the user within the authentication **domain** that is used to authenticate the user.

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

PictureUrl: The URL of a picture by which the user prefers to be recognized.

PictureTimestamp: A string that serves as a version identifier for the user's picture. The value of this property SHOULD change whenever the contents of PictureUrl and/or the contents of the picture stored at PictureUrl are altered.

PicturePlaceholderState: An integer that indicates whether the picture located at the URL stored in the PictureUrl property is a placeholder image. Valid values are "0" (not a placeholder) and "1" (placeholder).

PictureExchangeSyncState: An integer that indicates whether the picture located at the URL stored in the PictureUrl property was retrieved from Microsoft Exchange Server. Valid values are "0" (not retrieved from Microsoft Exchange Server) and "1" (retrieved from Microsoft Exchange Server).

PreferredName: The display name of the user by which the user prefers to be addressed.

ProfileSubtypeID: An integer that indicates the type of profile. Valid values are "1" (user) and "2" (organization).

RecordID: An 8-byte, user profile record identifier.

SID: A security identifier (SID) that uniquely identifies the user within the authentication domain that is used to authenticate the user.

SipAddress: The **Session Initiation Protocol (SIP) address** of the user.

Title: The work title of the user.

UserID: A 16-byte GUID that uniquely identifies the user.

PersonalSpace: The URL of the personal site for the user.

FeedIdentifier: A string that identifies the feed source for the user. The format of the string is feed application defined.

FeedPrivacyActivity: An integer that, if nonzero, indicates that changes to the user profile may generate feed activities.

IsPeopleListPublic: A Boolean that, if true, indicates that other users are permitted to obtain this users list of followed people.

EmailOptin: An integer field containing bitmask values that indicate the user-specified preferences for receiving emails related to feed activities. Emails are sent to the user under the following conditions:

- If the bit corresponding to value 0x08 is set, emails are sent to this user when a feed entry is posted to this user by any other feed user.
- If the bit corresponding to value 0x10 is set, emails are sent to this user when any other feed user replies to a feed entry posted by this user.
- If the bit corresponding to value 0x20 is set, emails are sent to this user when any other feed user replies to a reply posted by this user.

StatusNote: The body of the most recent non-reply feed entry generated by the user.

2.2.5 Simple Types

The following table summarizes the set of common XML schema simple type definitions defined by this specification. XML schema simple type definitions that are specific to a particular operation are described with the operation.

Simple type	Description
char	The **char** simple type is a **Unicode** character that is stored in a 4-byte integer.
duration	The **duration** simple type is the length of time, as specified in [XMLSCHEMA].
FaultCodes	The **FaultCodes** simple type is an enumeration of the possible errors that the service endpoint of this protocol communicates with protocol clients. The protocol imposes no restriction on the behavior of the protocol client with respect to this type.

2.2.5.1 char

Namespace: http://schemas.microsoft.com/2003/10/Serialization/

The **char** simple type is a Unicode character that is stored in a 4-byte integer.

```
<xs:simpleType name="char" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:restriction base="xs:int"/>
</xs:simpleType>
```

2.2.5.2 duration

Namespace: http://schemas.microsoft.com/2003/10/Serialization/

The **duration** simple type is the length of time, as specified in [XMLSCHEMA].

```
<xs:simpleType name="duration" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:restriction base="xs:duration">
    <xs:pattern value="\-?P(\d*D)?(T(\d*H)?(\d*M)?(\d*(\.\d*)?S)?)?"/>
    <xs:minInclusive value="-P10675199DT2H48M5.4775808S"/>
    <xs:maxInclusive value="P10675199DT2H48M5.4775807S"/>
  </xs:restriction>
</xs:simpleType>
```

2.2.5.3 FaultCodes

Namespace: http://Microsoft/Office/Server/UserProfiles

The **FaultCodes** simple type is an enumeration of the possible errors that the service endpoint of this protocol communicates with protocol clients. The protocol imposes no restriction on the behavior of the protocol client with respect to this type.

```
<xs:simpleType name="FaultCodes" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:restriction base="xs:string">
    <xs:enumeration value="UnknownError"/>
    <xs:enumeration value="InvalidInput"/>
```

```
      <xs:enumeration value="InvalidConfiguration"/>
    </xs:restriction>
  </xs:simpleType>
```

The following table specifies the allowable values for the **FaultCodes** simple type.

Value	Meaning
UnknownError	The cause of the error is unknown to the service endpoint of this protocol.
InvalidInput	The cause of the error is invalid **UserSearchCriteria** input from the protocol client.
InvalidConfiguration	The cause of the error is an invalid configuration of the service endpoint of this protocol or the transport layers below it.

2.2.6 Attributes

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

2.2.7 Groups

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

2.2.8 Attribute Groups

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

3 Protocol Details

In the following sections, the schema definition might differ from the processing rules imposed by the protocol. The WSDL in this specification matches the WSDL that shipped with the product and provides a base description of the schema. The text that introduces the WSDL might specify differences that reflect actual Microsoft product behavior. For example, the schema definition might allow for an element to be **empty**, **null**, or **not present** but the behavior of the protocol as specified restricts the same elements to being **non-empty**, **not null**, and **present**.

This protocol is a server-side protocol. The client side of this protocol is simply a pass-through. That is, no additional timers or other state is required on the client side of this protocol. Calls that are made by the upper protocol or application are passed directly to the transport, and the results returned by the transport are passed directly to the higher-layer protocol or application.

3.1 Server Details

This protocol is based on stateless interaction between the protocol client and protocol server. The protocol client MUST be authenticated with the credentials of a user who has access to the **User Profile Service** (referred to as 'UPA Standard Authentication' in the following figure). The protocol simply exposes the data that is currently stored in the User Profile Service that is deployed and running. There is no interdependency between the information exchanged between the protocol client and protocol server during one instance of their interaction using this protocol and the next instance. The following diagram illustrates the control flow for retrieving cached information about user profiles and returning it to the protocol client.

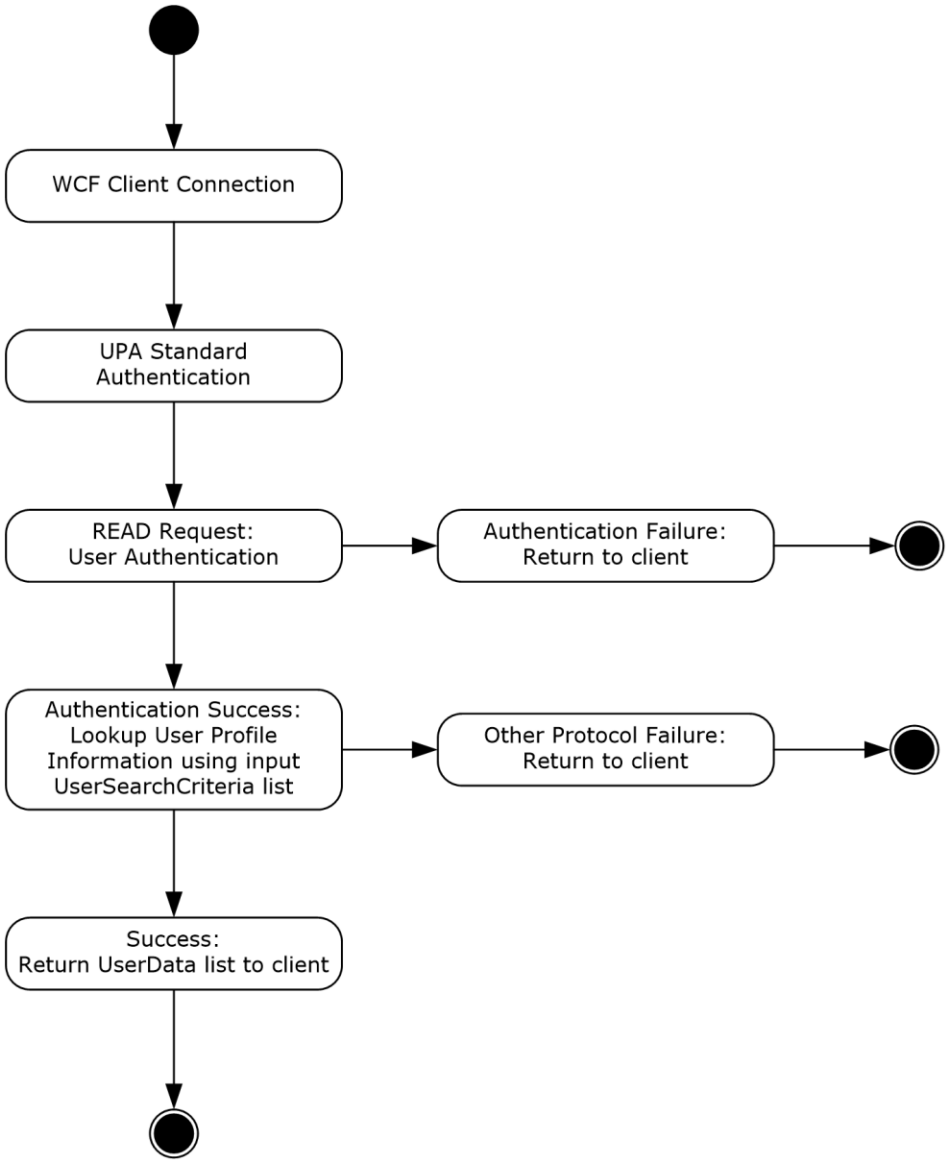

Figure 2: Control flow of the user profile caching service

When errors are returned to the protocol client, the service endpoint (4) of this protocol specifies complex types as described in sections 2.2.4.1 through 2.2.4.4 for each error.

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.

A key aspect of this protocol is that it allows protocol clients to gain quick and scalable access to a commonly used subset of User Profile Service properties and it does so through a WCF service endpoint that caches these properties from the back-end database server. Protocol clients can use the services that are offered by this protocol for commonly used user profile properties, such as email,

picture URL, user name, display name, and user profile record identifier. Protocol clients cannot use this protocol to update user profile information; this protocol does not support write requests.

Because this protocol does not support write requests, the cache of user profile properties behind the service endpoint of this protocol can potentially be out-of-date with the database layer before the cache is refreshed internally within the service. The following diagram captures these dependencies.

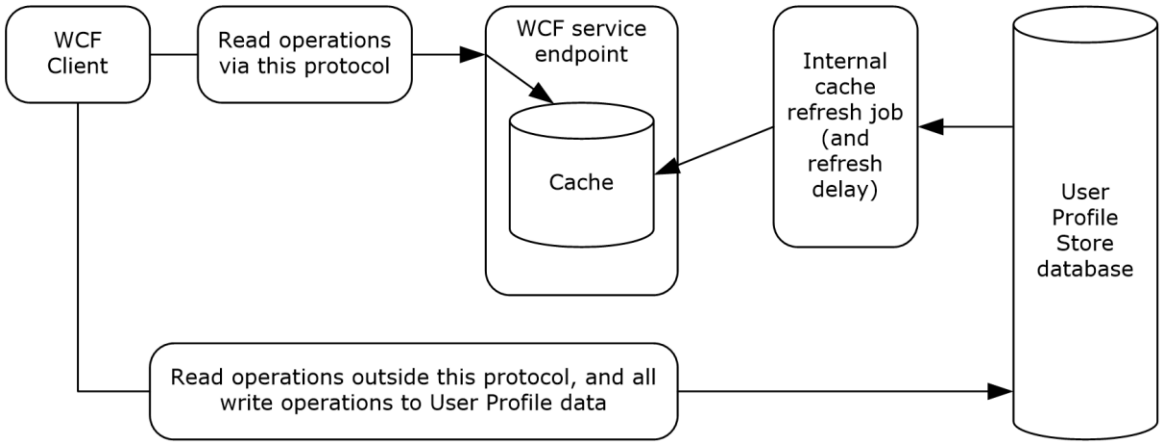

Figure 3: Dependencies on the cache

The preceding diagram illustrates that protocol clients that use this protocol MUST be able to tolerate a small delay, in the order of minutes, before a change that is made to the user profile data in the database is reflected in the cache. The cache refresh timer cannot be manipulated by using this protocol. It can be controlled only on the application server itself.

Initialization is not needed for the cache in the service endpoint because the cache is populated on demand and transparently whenever the protocol client accesses data that is missing in the cache but present in the database. The protocol client interface is a very simple stateless request/response protocol.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Message Processing Events and Sequencing Rules

The following table summarizes the list of operations as defined by this specification.

Operation	Description
GetUserData	The **GetUserData** operation retrieves the commonly used properties of a user profile in the user profile store.

3.1.4.1 GetUserData

The **GetUserData** operation retrieves the commonly used properties of a user profile in the user profile store.

The following is the WSDL port type specification of the **GetUserData** **WSDL operation**.

```
<wsdl:operation name="GetUserData" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input wsam:Action="http://Microsoft.Office.Server.UserProfiles/GetUserData"
message="tns:IProfileDBCacheService_GetUserData_InputMessage"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"/>
  <wsdl:output wsam:Action="http://tempuri.org/IProfileDBCacheService/GetUserDataResponse"
message="tns:IProfileDBCacheService GetUserData OutputMessage"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"/>
  <wsdl:fault
wsam:Action="http://tempuri.org/IProfileDBCacheService/GetUserDataConfigurationFaultFault"
name="ConfigurationFaultFault"
message="tns:IProfileDBCacheService_GetUserData_ConfigurationFaultFault_FaultMessage"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"/>
  <wsdl:fault
wsam:Action="http://tempuri.org/IProfileDBCacheService/GetUserDataInputFaultFault"
name="InputFaultFault"
message="tns:IProfileDBCacheService_GetUserData_InputFaultFault_FaultMessage"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"/>
  <wsdl:fault
wsam:Action="http://tempuri.org/IProfileDBCacheService/GetUserDataUnknownFaultFault"
name="UnknownFaultFault"
message="tns:IProfileDBCacheService_GetUserData_UnknownFaultFault_FaultMessage"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"/>
  <wsdl:fault
wsam:Action="http://tempuri.org/IProfileDBCacheService/GetUserDataUserProfileFaultFault"
name="UserProfileFaultFault"
message="tns:IProfileDBCacheService_GetUserData_UserProfileFaultFault_FaultMessage"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"/>
</wsdl:operation>
```

The protocol client sends an **IProfileDBCacheService_GetUserData_InputMessage** request message and the protocol server responds with an **IProfileDBCacheService_GetUserData_OutputMessage** response message. If the user profile store has user profile data that corresponds to the input **UserSearchCriteria** complex type, then this operation returns and packages that data for the protocol client as an output **UserData** complex type. The protocol client MAY send a request to retrieve multiple user profiles in one call to this operation, as specified in section 3.1.4.1.3.1. The operation returns one **UserData** complex type for each input that it receives if the operation finds the user data. Otherwise, the operation MUST return **NULL** in place of **UserData** corresponding to the **UserSearchCriteria**. In case of any error, the operation MUST return an error.If the operation returns a SOAP fault as specified in [SOAP1.1] section 4.4 or [SOAP1.2/1] section 5.4, the **SOAP fault detail** MUST contain one of the error complex types as specified in sections 2.2.4.1 through 2.2.4.4.

3.1.4.1.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
IProfileDBCacheService_GetUserData_InputMessage	The request WSDL message for the **GetUserData** WSDL operation.
IProfileDBCacheService_GetUserData_OutputMessage	The response WSDL message for the **GetUserData** WSDL operation.

3.1.4.1.1.1 IProfileDBCacheService_GetUserData_InputMessage

The request WSDL message for the **GetUserData** WSDL operation.

The **SOAP action** value is:

```
http://Microsoft.Office.Server.UserProfiles/GetUserData
```

The **SOAP body** contains the **GetUserData** element.

3.1.4.1.1.2 IProfileDBCacheService_GetUserData_OutputMessage

The response WSDL message for the **GetUserData** WSDL operation.

The SOAP body contains the **GetUserDataResponse** element.

3.1.4.1.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
GetUserData	The input data for the **GetUserData** WSDL operation.
GetUserDataResponse	The result data for the **GetUserData** WSDL operation.

3.1.4.1.2.1 GetUserData

The **GetUserData** element specifies the input data for the **GetUserData** WSDL operation.

```
<xs:element name="GetUserData" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element xmlns:q1="http://Microsoft/Office/Server/UserProfiles" minOccurs="0"
name="searchCriteria" nillable="true" type="q1:UserSearchCriteria"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

searchCriteria: Contains search criteria for the service endpoint to use when searching for user profile data.

3.1.4.1.2.2 GetUserDataResponse

The **GetUserDataResponse** element specifies the result data for the **GetUserData** WSDL operation.

```
<xs:element name="GetUserDataResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element
xmlns:q2="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.UserProfiles.Cache"
minOccurs="0" name="GetUserDataResult" nillable="true"
type="q2:StreamedDataOfArrayOfUserDatajHCugpoN"/>
    </xs:sequence>
```

```
    </xs:complexType>
  </xs:element>
```

GetUserDataResult: Contains **StreamedDataOfArrayOfUserDatajHCugpoN** complex type. The collection of **UserData** complex types specified in **StreamedDataOfArrayOfUserDatajHCugpoN** complex type have a one-to-one relationship with input **searchCriteria** complex types. If the user profile data for a specific **UserSearchCriteria** is not found, the corresponding **UserData** structure is nil.

3.1.4.1.3 Complex Types

The following table summarizes the XML schema complex type definitions that are specific to this operation.

Complex type	Description
ArrayOfbase64Binary	**ArrayOfbase64Binary** is a sequence of zero or more **base64Binary** types.
ArrayOfguid	**ArrayOfguid** is a sequence of zero or more 16-byte GUIDs.
ArrayOflong	**ArrayOflong** is a sequence of zero or more 8-byte, long integers.
ArrayOfstring	**ArrayOfstring** is a sequence of zero or more strings.
MarshalByRefObject	**MarshalByRefObject** is the base class for objects that communicate across application domain boundaries by exchanging messages using a proxy.
MemoryStream	**MemoryStream** is a complex type that specifies a **stream** whose backing store is memory.
Stream	**Stream** is a complex type that specifies a stream (1).
StreamedDataOfArrayOfUserDatajHCugpoN	**StreamedDataOfArrayOfUserDatajHCugpoN** is a complex type that contains a stream with serialized contents of **ArrayOfUserData** complex type.
UserSearchCriteria	The **UserSearchCriteria** complex type specifies the list of properties that the service endpoint MUST use when searching for user profile data. In case of any error, the protocol server MUST return either an **InputFault**, **ConfigurationFault**, or **UnknownFault** complex type to the protocol client.

3.1.4.1.3.1 UserSearchCriteria

Namespace: http://Microsoft/Office/Server/UserProfiles

The **UserSearchCriteria** complex type specifies the list of properties that the service endpoint MUST use when searching for user profile data. In case of any error, the protocol server MUST return either an **InputFault**, **ConfigurationFault**, or **UnknownFault** complex type to the protocol client.

```
<xs:complexType name="UserSearchCriteria" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" name="AllowAlternateAccountName" type="xs:boolean"/>
    <xs:element minOccurs="0" name="CorrelationID" type="tns1:guid"/>
```

```
    <xs:element xmlns:q3="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
minOccurs="0" name="EmailCollection" nillable="true" type="q3:ArrayOfstring"/>
    <xs:element xmlns:q3="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
minOccurs="0" name="NTNameCollection" nillable="true" type="q3:ArrayOfstring"/>
    <xs:element minOccurs="0" name="PartitionID" type="tns1:guid"/>
    <xs:element xmlns:q3="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
minOccurs="0" name="RecordIDCollection" nillable="true" type="q3:ArrayOflong"/>
    <xs:element xmlns:q3="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
minOccurs="0" name="SIDCollection" nillable="true" type="q3:ArrayOfbase64Binary"/>
    <xs:element minOccurs="0" name="SearchColumn" nillable="true" type="xs:string"/>
    <xs:element xmlns:q3="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
minOccurs="0" name="UserIDCollection" nillable="true" type="q3:ArrayOfguid"/>
  </xs:sequence>
</xs:complexType>
```

AllowAlternateAccountName: A single user can have multiple accounts in an organization. For example, a user can have an account "DOMAIN1\user1" for primary **authentication** and another account "DOMAIN2\User1First.User1Last" for email. When a user's user profile information is created, a distinct record for each user account is created in the user profile store, but the service ensures that all of the accounts have the same user profile data associated with them in the user profile store. The service also marks one of the accounts, for example "DOMAIN1\user1" as the master account, which is used primarily to identify the user. If this value is "true", it indicates that the **UserSearchCriteria** can be applied to any of the user's accounts. If this value is "false", it indicates that **UserSearchCriteria** MUST be applied only to the master account for the user.

CorrelationID: The optional **request identifier** for the current request.

EmailCollection: The list of email addresses for users, based on which the search for their user profile data MUST be conducted by the service. A commonly used format is "user@domain.company".

NTNameCollection: The list of user names for users, based on which the search for their user profile data MUST be conducted by the service. A commonly used format is "DOMAIN\user".

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

RecordIDCollection: The list of 8-byte user profile record identifiers, based on which the search for the user profile data MUST be conducted by the service.

SIDCollection: The list of security identifiers (SIDs), based on which the search for the user profile data MUST be conducted by the service.

SearchColumn: Specifies which of the following search criteria fields to use when searching for user profile data:

- If the **SearchColumn** is the string "Email" it specifies **EmailCollection**.
- If the **SearchColumn** is the string "UserID" it specifies **UserIDCollection**.
- If the **SearchColumn** is the string "NTName" it specifies **NTNameCollection**.
- If the **SearchColumn** is the string "SID" it specifies **SIDCollection**.
- If the **SearchColumn** is the string "RecordID" it specifies **RecordIDCollection**.

  The service never matches against more than one of these fields at a time. The service chooses exactly one string specified by **SearchColumn** to conduct a search. This value MUST NOT be null or empty.

UserIDCollection: The list of 16-byte GUIDs that uniquely identify a user, based on which the search for the user profile data MUST be conducted by the service.

3.1.4.1.3.2 ArrayOfstring

Namespace: http://schemas.microsoft.com/2003/10/Serialization/Arrays

ArrayOfstring is a sequence of zero or more strings.

```
<xs:complexType name="ArrayOfstring" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="string" nillable="true"
type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

string: A single string value.

3.1.4.1.3.3 ArrayOflong

Namespace: http://schemas.microsoft.com/2003/10/Serialization/Arrays

ArrayOflong is a sequence of zero or more 8-byte, long integers.

```
<xs:complexType name="ArrayOflong" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="long" type="xs:long"/>
  </xs:sequence>
</xs:complexType>
```

long: An 8-byte (64-bit) integer.

3.1.4.1.3.4 ArrayOfbase64Binary

Namespace: http://schemas.microsoft.com/2003/10/Serialization/Arrays

ArrayOfbase64Binary is a sequence of zero or more **base64Binary** types.

```
<xs:complexType name="ArrayOfbase64Binary" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="base64Binary" nillable="true"
type="xs:base64Binary"/>
  </xs:sequence>
</xs:complexType>
```

base64Binary: This is a sequence of bytes represented in **base64 encoding**.

3.1.4.1.3.5 ArrayOfguid

Namespace: http://schemas.microsoft.com/2003/10/Serialization/Arrays

ArrayOfguid is a sequence of zero or more 16-byte GUIDs.

```
<xs:complexType name="ArrayOfguid" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="guid" type="tns1:guid"/>
  </xs:sequence>
</xs:complexType>
```

guid: A 16-byte GUID.

3.1.4.1.3.6 StreamedDataOfArrayOfUserDatajHCugpoN

Namespace: http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.UserProfiles.Cache

StreamedDataOfArrayOfUserDatajHCugpoN is a complex type that contains a stream with serialized contents of **ArrayOfUserData** complex type**.**

```
<xs:complexType name="StreamedDataOfArrayOfUserDatajHCugpoN"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element xmlns:q4="http://schemas.datacontract.org/2004/07/System.IO" minOccurs="0"
name="DataStream" nillable="true" type="q4:MemoryStream"/>
  </xs:sequence>
</xs:complexType>
```

DataStream: Specifies a **MemoryStream** complex type containing serialized contents of **ArrayOfUserData** complex type.

3.1.4.1.3.7 MemoryStream

Namespace: http://schemas.datacontract.org/2004/07/System.IO

MemoryStream is a complex type that specifies a stream whose backing store is memory.

```
<xs:complexType name="MemoryStream" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexContent mixed="false">
    <xs:extension base="q4:Stream">
      <xs:sequence>
        <xs:element name="_buffer" nillable="true" type="xs:base64Binary"/>
        <xs:element name="_capacity" type="xs:int"/>
        <xs:element name="_expandable" type="xs:boolean"/>
        <xs:element name="_exposable" type="xs:boolean"/>
        <xs:element name="_isOpen" type="xs:boolean"/>
        <xs:element name="_length" type="xs:int"/>
        <xs:element name="_origin" type="xs:int"/>
        <xs:element name="_position" type="xs:int"/>
        <xs:element name="_writable" type="xs:boolean"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

_buffer: The array of unsigned bytes from which the stream is created. It represents serialized contents of **ArrayOfUserData** complex type.

_capacity: Specifies the number of bytes allocated for this stream.

_expandable: Specifies if the stream is expandable.

_exposable: Specifies if the underlying byte buffer is exposed.

_isOpen: Specifies if the stream is open.

_length: Specifies the length of the stream in bytes.

_origin: Specifies the index into the underlying buffer at which the stream starts.

_position: Specifies the current position within the stream.

_writable: Specifies if the stream supports writing.

3.1.4.1.3.8 Stream

Namespace: http://schemas.datacontract.org/2004/07/System.IO

Stream is a complex type that specifies a stream.

```
<xs:complexType name="Stream" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexContent mixed="false">
    <xs:extension xmlns:q5="http://schemas.datacontract.org/2004/07/System"
base="q5:MarshalByRefObject">
      <xs:sequence/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

3.1.4.1.3.9 MarshalByRefObject

Namespace: http://schemas.datacontract.org/2004/07/System

MarshalByRefObject is the base class for objects that communicate across application domain boundaries by exchanging messages using a proxy.

```
<xs:complexType name="MarshalByRefObject" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element name="__identity" nillable="true" type="xs:anyType"/>
  </xs:sequence>
</xs:complexType>
```

__identity: Holds marshalling information of the object.

3.1.4.1.4 Simple Types

The following table summarizes the XML schema simple type definitions that are specific to this operation.

Simple type	Description
guid	**guid** is a 16-byte GUID.

3.1.4.1.4.1 guid

Namespace: http://schemas.microsoft.com/2003/10/Serialization/

guid is a 16-byte GUID.

```
<xs:simpleType name="guid" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:restriction base="xs:string">
    <xs:pattern value="[\da-fA-F]{8}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-
F]{12}"/>
  </xs:restriction>
</xs:simpleType>
```

3.1.4.1.5 Attributes

None.

3.1.4.1.6 Groups

None.

3.1.4.1.7 Attribute Groups

None.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4 Protocol Examples

This section provides specific example scenarios for retrieving a single user profile by **login name** and retrieving multiple user profiles by user profile record identifier.

Consider that the following two user profiles are part of the user profile store:

Record ID	User Name	NT Name	User ID
1	Agarwal, Nupur	contoso\nupuragarwal	4bccfd46-fb01-4c9b-988c-0730eaed529a
2	Cannon, Paul	contoso\paulcannon	b8110f57-8f8e-4c4a-9570-1c75828e18ef

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

- PartitionID - 0C37852B-34D0-418E-91C6-2AC25AF4BE5B
- CorrelationID - a6b889ea-932a-4447-8d73-ce928cc912ad
- Security identifier (SID) - AQUAAAAAAAUVAAAAoGXPfnhLm1/nfIdw6iooAA==

4.1 Retrieving a User Profile by NT Name

To retrieve a user profile by domain user account, consider the following WSDL message constructed by the protocol client to call **GetUserData.**

```
  <s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
    <s:Header>
      <a:Action
s:mustUnderstand="1">http://Microsoft.Office.Server.UserProfiles/GetUserData</a:Action>
      <a:MessageID>urn:uuid:3548f307-1764-4099-bbab-d421557cca8d</a:MessageID>
      <a:ReplyTo>
        <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
      </a:ReplyTo>
      <a:To
s:mustUnderstand="1">net.tcp://mymachine:32845/827f1817ac334b5da4476147b9ee5a8f/ProfileDBCach
eService.svc</a:To>
      <o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-secext-1.0.xsd">
        ...
      </o:Security>
    </s:Header>
    <s:Body>
      <GetUserData xmlns="http://tempuri.org/">
        <searchCriteria xmlns:b="http://Microsoft/Office/Server/UserProfiles"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
          <b:AllowAlternateAccountName>false</b:AllowAlternateAccountName>
          <b:CorrelationID>a6b889ea-932a-4447-8d73-ce928cc912ad</b:CorrelationID>
          <b:EmailCollection i:nil="true"
xmlns:c="http://schemas.microsoft.com/2003/10/Serialization/Arrays"></b:EmailCollection>
          <b:NTNameCollection
xmlns:c="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
            <c:string>contoso\nupuragarwal</c:string>
          </b:NTNameCollection>
          <b:PartitionID>0c37852b-34d0-418e-91c6-2ac25af4be5b</b:PartitionID>
          <b:RecordIDCollection i:nil="true"
xmlns:c="http://schemas.microsoft.com/2003/10/Serialization/Arrays"></b:RecordIDCollection>
          <b:SIDCollection i:nil="true"
xmlns:c="http://schemas.microsoft.com/2003/10/Serialization/Arrays"></b:SIDCollection>
          <b:SearchColumn>NTName</b:SearchColumn>
```

```
          <b:UserIDCollection i:nil="true"
xmlns:c="http://schemas.microsoft.com/2003/10/Serialization/Arrays"></b:UserIDCollection>
        </searchCriteria>
      </GetUserData>
    </s:Body>
  </s:Envelope>
```

The protocol server returns the following WSDL message response:

```
  <s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
    <s:Header>
      <a:Action
s:mustUnderstand="1">http://tempuri.org/IProfileDBCacheService/GetUserDataResponse</a:Action>
      <a:RelatesTo>urn:uuid:3548f307-1764-4099-bbab-d421557cca8d</a:RelatesTo>
      <a:To s:mustUnderstand="1">http://www.w3.org/2005/08/addressing/anonymous</a:To>
      <o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-secext-1.0.xsd">
        ...
      </o:Security>
    </s:Header>
    <s:Body>
      <GetUserDataResponse xmlns="http://tempuri.org/">
        <GetUserDataResult xmlns:b="http://Microsoft/Office/Server/UserProfiles"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
          <d4p1:DataStream xmlns:d5p1="http://schemas.datacontract.org/2004/07/System.IO">
            <__identity i:nil="true"
xmlns="http://schemas.datacontract.org/2004/07/System"></  identity>
            <d5p1:_buffer>…</d5p1:_buffer>
            <d5p1: capacity>672</d5p1: capacity>
            <d5p1:_expandable>true</d5p1:_expandable>
            <d5p1:_exposable>true</d5p1:_exposable>
            <d5p1:_isOpen>true</d5p1:_isOpen>
            <d5p1:_length>672</d5p1:_length>
            <d5p1:_origin>0</d5p1:_origin>
            <d5p1: position>0</d5p1: position>
            <d5p1: writable>true</d5p1: writable>
          </d4p1:DataStream>
        </GetUserDataResult>
      </GetUserDataResponse>
    </s:Body>
  </s:Envelope>
```

4.2 Retrieving Multiple User Profiles by Record Identifier

To retrieve multiple user profiles by record identifier, consider the following WSDL message constructed by the protocol client to call **GetUserData**.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  <s:Header>
    <a:Action
s:mustUnderstand="1">http://Microsoft.Office.Server.UserProfiles/GetUserData</a:Action>
    <a:MessageID>urn:uuid:d0d7a499-931c-4e16-8407-6ea7f43b2a4a</a:MessageID>
    <a:ReplyTo>
      <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
    </a:ReplyTo>
    <a:To
s:mustUnderstand="1">net.tcp://mymachine:32845/827f1817ac334b5da4476147b9ee5a8f/ProfileDBCach
eService.svc</a:To>
    <o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-secext-1.0.xsd">
      ...
    </o:Security>
```

```
  </s:Header>
  <s:Body>
    <GetUserData xmlns="http://tempuri.org/">
      <searchCriteria xmlns:b="http://Microsoft/Office/Server/UserProfiles"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <b:AllowAlternateAccountName>true</b:AllowAlternateAccountName>
        <b:CorrelationID>a6b889ea-932a-4447-8d73-ce928cc912ad</b:CorrelationID>
        <b:EmailCollection i:nil="true"
xmlns:c="http://schemas.microsoft.com/2003/10/Serialization/Arrays"></b:EmailCollection>
        <b:NTNameCollection i:nil="true"
xmlns:c="http://schemas.microsoft.com/2003/10/Serialization/Arrays"></b:NTNameCollection>
        <b:PartitionID>0c37852b-34d0-418e-91c6-2ac25af4be5b</b:PartitionID>
        <b:RecordIDCollection xmlns:c="http://www.w3.org/2001/XMLSchema">
          <c:long>1</c:long>
          <c:long>2</c:long>
        </b:RecordIDCollection>
        <b:SIDCollection i:nil="true"
xmlns:c="http://schemas.microsoft.com/2003/10/Serialization/Arrays"></b:SIDCollection>
        <b:SearchColumn>RecordID</b:SearchColumn>
        <b:UserIDCollection i:nil="true"
xmlns:c="http://schemas.microsoft.com/2003/10/Serialization/Arrays"></b:UserIDCollection>
      </searchCriteria>
    </GetUserData>
  </s:Body>
</s:Envelope>
```

The protocol server responds with the following WSDL message:

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  <s:Header>
    <a:Action
s:mustUnderstand="1">http://tempuri.org/IProfileDBCacheService/GetUserDataResponse</a:Action>
    <a:RelatesTo>urn:uuid:d0d7a499-931c-4e16-8407-6ea7f43b2a4a</a:RelatesTo>
    <a:To s:mustUnderstand="1">http://www.w3.org/2005/08/addressing/anonymous</a:To>
    <o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-secext-1.0.xsd">
      ...
    </o:Security>
  </s:Header>
  <s:Body>
    <GetUserDataResponse xmlns="http://tempuri.org/">
      <GetUserDataResult xmlns:b="http://Microsoft/Office/Server/UserProfiles"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <d4p1:DataStream xmlns:d5p1="http://schemas.datacontract.org/2004/07/System.IO">
          <__identity i:nil="true"
xmlns="http://schemas.datacontract.org/2004/07/System"></__identity>
          <d5p1: buffer>…</d5p1: buffer>
          <d5p1:_capacity>2054</d5p1:_capacity>
          <d5p1:_expandable>true</d5p1:_expandable>
          <d5p1:_exposable>true</d5p1:_exposable>
          <d5p1:_isOpen>true</d5p1:_isOpen>
          <d5p1: length>1205</d5p1: length>
          <d5p1:_origin>0</d5p1:_origin>
          <d5p1:_position>0</d5p1:_position>
          <d5p1:_writable>true</d5p1:_writable>
        </d4p1:DataStream>
      </GetUserDataResult>
    </GetUserDataResponse>
  </s:Body>
</s:Envelope>
```

5 Security

5.1 Security Considerations for Implementers

There are no security considerations that are specific to this protocol. General security considerations pertaining to [RFC2822] apply.

This protocol does not introduce any additional security considerations beyond those that apply to its underlying protocols.

5.2 Index of Security Parameters

None.

6 Appendix A: Full WSDL

For ease of implementation, the full WSDL is provided in this appendix.

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://tempuri.org/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata" name="ProfileDBCacheService"
targetNamespace="http://tempuri.org/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <xs:schema xmlns:tns2="http://tempuri.org/Imports"
targetNamespace="http://tempuri.org/Imports">
      <xs:import namespace="http://Microsoft/Office/Server/UserProfiles"/>
      <xs:import
namespace="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.UserProfiles.Cache
"/>
      <xs:import namespace="http://schemas.datacontract.org/2004/07/System"/>
      <xs:import namespace="http://schemas.datacontract.org/2004/07/System.IO"/>
      <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/"/>
      <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
      <xs:import namespace="http://tempuri.org/"/>
    </xs:schema>
  </wsdl:types>
  <wsdl:portType name="IProfileDBCacheService">
    <wsdl:operation name="GetUserData">
      <wsdl:input wsam:Action="http://Microsoft.Office.Server.UserProfiles/GetUserData"
message="tns:IProfileDBCacheService_GetUserData_InputMessage"/>
      <wsdl:output
wsam:Action="http://tempuri.org/IProfileDBCacheService/GetUserDataResponse"
message="tns:IProfileDBCacheService_GetUserData_OutputMessage"/>
      <wsdl:fault
wsam:Action="http://tempuri.org/IProfileDBCacheService/GetUserDataConfigurationFaultFault"
name="ConfigurationFaultFault"
message="tns:IProfileDBCacheService_GetUserData_ConfigurationFaultFault_FaultMessage"/>
      <wsdl:fault
wsam:Action="http://tempuri.org/IProfileDBCacheService/GetUserDataInputFaultFault"
name="InputFaultFault"
message="tns:IProfileDBCacheService_GetUserData_InputFaultFault_FaultMessage"/>
      <wsdl:fault
wsam:Action="http://tempuri.org/IProfileDBCacheService/GetUserDataUnknownFaultFault"
name="UnknownFaultFault"
message="tns:IProfileDBCacheService_GetUserData_UnknownFaultFault_FaultMessage"/>
      <wsdl:fault
wsam:Action="http://tempuri.org/IProfileDBCacheService/GetUserDataUserProfileFaultFault"
name="UserProfileFaultFault"
message="tns:IProfileDBCacheService GetUserData UserProfileFaultFault FaultMessage"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="CustomBinding_IProfileDBCacheService"
type="tns:IProfileDBCacheService">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="GetUserData">
      <soap:operation soapAction="http://Microsoft.Office.Server.UserProfiles/GetUserData"
style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="ConfigurationFaultFault">
        <soap:fault use="literal" name="ConfigurationFaultFault" namespace=""/>
      </wsdl:fault>
      <wsdl:fault name="InputFaultFault">
        <soap:fault use="literal" name="InputFaultFault" namespace=""/>
      </wsdl:fault>
      <wsdl:fault name="UnknownFaultFault">
```

```
        <soap:fault use="literal" name="UnknownFaultFault" namespace=""/>
      </wsdl:fault>
      <wsdl:fault name="UserProfileFaultFault">
        <soap:fault use="literal" name="UserProfileFaultFault" namespace=""/>
      </wsdl:fault>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:message
name="IProfileDBCacheService_GetUserData_ConfigurationFaultFault_FaultMessage">
    <wsdl:part xmlns:q1="http://Microsoft/Office/Server/UserProfiles" name="detail"
element="q1:ConfigurationFault"/>
  </wsdl:message>
  <wsdl:message name="IProfileDBCacheService_GetUserData_InputFaultFault_FaultMessage">
    <wsdl:part xmlns:q1="http://Microsoft/Office/Server/UserProfiles" name="detail"
element="q1:InputFault"/>
  </wsdl:message>
  <wsdl:message name="IProfileDBCacheService GetUserData InputMessage">
    <wsdl:part name="parameters" element="tns:GetUserData"/>
  </wsdl:message>
  <wsdl:message name="IProfileDBCacheService_GetUserData_OutputMessage">
    <wsdl:part name="parameters" element="tns:GetUserDataResponse"/>
  </wsdl:message>
  <wsdl:message name="IProfileDBCacheService GetUserData UnknownFaultFault FaultMessage">
    <wsdl:part xmlns:q1="http://Microsoft/Office/Server/UserProfiles" name="detail"
element="q1:UnknownFault"/>
  </wsdl:message>
  <wsdl:message name="IProfileDBCacheService_GetUserData_UserProfileFaultFault_FaultMessage">
    <wsdl:part xmlns:q1="http://Microsoft/Office/Server/UserProfiles" name="detail"
element="q1:UserProfileFault"/>
  </wsdl:message>
</wsdl:definitions>
```

7 Appendix B: Full XML Schema

Schema name	Prefix	Section
http://tempuri.org/	tns	7.1
http://schemas.microsoft.com/2003/10/Serialization/	tns1	7.2
http://Microsoft/Office/Server/UserProfiles	q1	7.3
http://schemas.microsoft.com/2003/10/Serialization/Arrays	q3	7.4
http://schemas.datacontract.org/2004/07 /Microsoft.Office.Server.UserProfiles.Cache	q2	7.5
http://schemas.datacontract.org/2004/07/System.IO	q4	7.6
http://schemas.datacontract.org/2004/07/System	q5	7.7

For ease of implementation, the following sections provide the full XML schema for this protocol.

7.1 http://tempuri.org/ Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="GetUserData">
    <xs:complexType>
      <xs:sequence>
        <xs:element xmlns:q1="http://Microsoft/Office/Server/UserProfiles" minOccurs="0"
name="searchCriteria" nillable="true" type="q1:UserSearchCriteria"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="GetUserDataResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element
xmlns:q2="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.UserProfiles.Cache"
minOccurs="0" name="GetUserDataResult" nillable="true"
type="q2:StreamedDataOfArrayOfUserDatajHCugpoN"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:import namespace="http://Microsoft/Office/Server/UserProfiles"/>
  <xs:import
namespace="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.UserProfiles.Cache
"/>
</xs:schema>
```

7.2 http://schemas.microsoft.com/2003/10/Serialization/ Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:tns1="http://schemas.microsoft.com/2003/10/Serialization/"
attributeFormDefault="qualified" elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/2003/10/Serialization/"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="anyType" nillable="true" type="xs:anyType"/>
  <xs:element name="anyURI" nillable="true" type="xs:anyURI"/>
  <xs:element name="base64Binary" nillable="true" type="xs:base64Binary"/>
  <xs:element name="boolean" nillable="true" type="xs:boolean"/>
```

```
  <xs:element name="byte" nillable="true" type="xs:byte"/>
  <xs:element name="dateTime" nillable="true" type="xs:dateTime"/>
  <xs:element name="decimal" nillable="true" type="xs:decimal"/>
  <xs:element name="double" nillable="true" type="xs:double"/>
  <xs:element name="float" nillable="true" type="xs:float"/>
  <xs:element name="int" nillable="true" type="xs:int"/>
  <xs:element name="long" nillable="true" type="xs:long"/>
  <xs:element name="QName" nillable="true" type="xs:QName"/>
  <xs:element name="short" nillable="true" type="xs:short"/>
  <xs:element name="string" nillable="true" type="xs:string"/>
  <xs:element name="unsignedByte" nillable="true" type="xs:unsignedByte"/>
  <xs:element name="unsignedInt" nillable="true" type="xs:unsignedInt"/>
  <xs:element name="unsignedLong" nillable="true" type="xs:unsignedLong"/>
  <xs:element name="unsignedShort" nillable="true" type="xs:unsignedShort"/>
  <xs:element name="char" nillable="true" type="tns1:char"/>
  <xs:simpleType name="char">
    <xs:restriction base="xs:int"/>
  </xs:simpleType>
  <xs:element name="duration" nillable="true" type="tns1:duration"/>
  <xs:simpleType name="duration">
    <xs:restriction base="xs:duration">
      <xs:pattern value="\-?P(\d*D)?(T(\d*H)?(\d*M)?(\d*(\.\d*)?S)?)?"/>
      <xs:minInclusive value="-P10675199DT2H48M5.4775808S"/>
      <xs:maxInclusive value="P10675199DT2H48M5.4775807S"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="guid" nillable="true" type="tns1:guid"/>
  <xs:simpleType name="guid">
    <xs:restriction base="xs:string">
      <xs:pattern value="[\da-fA-F]{8}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-
F]{12}"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:attribute name="FactoryType" type="xs:QName"/>
  <xs:attribute name="Id" type="xs:ID"/>
  <xs:attribute name="Ref" type="xs:IDREF"/>
</xs:schema>
```

7.3 http://Microsoft/Office/Server/UserProfiles Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:q1="http://Microsoft/Office/Server/UserProfiles"
xmlns:tns1="http://schemas.microsoft.com/2003/10/Serialization/"
elementFormDefault="qualified" targetNamespace="http://Microsoft/Office/Server/UserProfiles"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="UserSearchCriteria">
    <xs:sequence>
      <xs:element minOccurs="0" name="AllowAlternateAccountName" type="xs:boolean"/>
      <xs:element minOccurs="0" name="CorrelationID" type="tns1:guid"/>
      <xs:element xmlns:q3="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
minOccurs="0" name="EmailCollection" nillable="true" type="q3:ArrayOfstring"/>
      <xs:element xmlns:q3="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
minOccurs="0" name="NTNameCollection" nillable="true" type="q3:ArrayOfstring"/>
      <xs:element minOccurs="0" name="PartitionID" type="tns1:guid"/>
      <xs:element xmlns:q3="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
minOccurs="0" name="RecordIDCollection" nillable="true" type="q3:ArrayOflong"/>
      <xs:element xmlns:q3="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
minOccurs="0" name="SIDCollection" nillable="true" type="q3:ArrayOfbase64Binary"/>
      <xs:element minOccurs="0" name="SearchColumn" nillable="true" type="xs:string"/>
      <xs:element xmlns:q3="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
minOccurs="0" name="UserIDCollection" nillable="true" type="q3:ArrayOfguid"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="UserSearchCriteria" nillable="true" type="q1:UserSearchCriteria"/>
  <xs:complexType name="UnknownFault">
    <xs:complexContent mixed="false">
      <xs:extension base="q1:UserProfileFault">
```

```
        <xs:sequence/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="UnknownFault" nillable="true" type="q1:UnknownFault"/>
  <xs:complexType name="UserProfileFault">
    <xs:sequence>
      <xs:element minOccurs="0" name="FailureDetail" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="FaultCode" type="q1:FaultCodes"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="UserProfileFault" nillable="true" type="q1:UserProfileFault"/>
  <xs:simpleType name="FaultCodes">
    <xs:restriction base="xs:string">
      <xs:enumeration value="UnknownError"/>
      <xs:enumeration value="InvalidInput"/>
      <xs:enumeration value="InvalidConfiguration"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="FaultCodes" nillable="true" type="q1:FaultCodes"/>
  <xs:complexType name="ConfigurationFault">
    <xs:complexContent mixed="false">
      <xs:extension base="q1:UserProfileFault">
        <xs:sequence/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="ConfigurationFault" nillable="true" type="q1:ConfigurationFault"/>
  <xs:complexType name="InputFault">
    <xs:complexContent mixed="false">
      <xs:extension base="q1:UserProfileFault">
        <xs:sequence/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="InputFault" nillable="true" type="q1:InputFault"/>
  <xs:complexType name="ArrayOfUserData">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="UserData" nillable="true"
type="q1:UserData"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="ArrayOfUserData" nillable="true" type="q1:ArrayOfUserData"/>
  <xs:complexType name="UserData">
    <xs:sequence>
      <xs:element minOccurs="0" name="Department" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="Email" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="LastUpdate" type="xs:dateTime"/>
      <xs:element minOccurs="0" name="MasterRecordID" type="xs:long"/>
      <xs:element minOccurs="0" name="NTName" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="PartitionID" type="tns1:guid"/>
      <xs:element minOccurs="0" name="PictureUrl" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="PreferredName" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="ProfileSubtypeID" type="xs:int"/>
      <xs:element minOccurs="0" name="RecordID" type="xs:long"/>
      <xs:element minOccurs="0" name="SID" nillable="true" type="xs:base64Binary"/>
      <xs:element minOccurs="0" name="SipAddress" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="Title" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="UserID" type="tns1:guid"/>
      <xs:element minOccurs="0" name="PersonalSpace" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="FeedIdentifier" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="FeedPrivacyActivity" nillable="true" type="xs:int"/>
      <xs:element minOccurs="0" name="IsPeopleListPublic" nillable="true" type="xs:boolean"/>
      <xs:element minOccurs="0" name="EmailOptin" nillable="true" type="xs:int"/>
      <xs:element minOccurs="0" name="StatusNote" nillable="true" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="UserData" nillable="true" type="q1:UserData"/>
  <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/"/>
```

```
  <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
</xs:schema>
```

7.4 http://schemas.microsoft.com/2003/10/Serialization/Arrays Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:tns1="http://schemas.microsoft.com/2003/10/Serialization/"
xmlns:q3="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/"/>
  <xs:complexType name="ArrayOfstring">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="string" nillable="true"
type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="ArrayOfstring" nillable="true" type="q3:ArrayOfstring"/>
  <xs:complexType name="ArrayOflong">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="long" type="xs:long"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="ArrayOflong" nillable="true" type="q3:ArrayOflong"/>
  <xs:complexType name="ArrayOfbase64Binary">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="base64Binary" nillable="true"
type="xs:base64Binary"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="ArrayOfbase64Binary" nillable="true" type="q3:ArrayOfbase64Binary"/>
  <xs:complexType name="ArrayOfguid">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="guid" type="tns1:guid"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="ArrayOfguid" nillable="true" type="q3:ArrayOfguid"/>
</xs:schema>
```

7.5 http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.UserProfiles.Cache Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
xmlns:q2="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.UserProfiles.Cache"
elementFormDefault="qualified"
targetNamespace="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.UserProfiles
.Cache" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:import namespace="http://schemas.datacontract.org/2004/07/System.IO"/>
  <xs:complexType name="StreamedDataOfArrayOfUserDatajHCugpoN">
    <xs:sequence>
      <xs:element xmlns:q4="http://schemas.datacontract.org/2004/07/System.IO" minOccurs="0"
name="DataStream" nillable="true" type="q4:MemoryStream"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="StreamedDataOfArrayOfUserDatajHCugpoN" nillable="true"
type="q2:StreamedDataOfArrayOfUserDatajHCugpoN"/>
</xs:schema>
```

7.6 http://schemas.datacontract.org/2004/07/System.IO Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:q4="http://schemas.datacontract.org/2004/07/System.IO"
elementFormDefault="qualified"
targetNamespace="http://schemas.datacontract.org/2004/07/System.IO"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:import namespace="http://schemas.datacontract.org/2004/07/System"/>
  <xs:complexType name="MemoryStream">
    <xs:complexContent mixed="false">
      <xs:extension base="q4:Stream">
        <xs:sequence>
          <xs:element name="_buffer" nillable="true" type="xs:base64Binary"/>
          <xs:element name="_capacity" type="xs:int"/>
          <xs:element name="_expandable" type="xs:boolean"/>
          <xs:element name="_exposable" type="xs:boolean"/>
          <xs:element name="_isOpen" type="xs:boolean"/>
          <xs:element name="_length" type="xs:int"/>
          <xs:element name="_origin" type="xs:int"/>
          <xs:element name="_position" type="xs:int"/>
          <xs:element name="_writable" type="xs:boolean"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="MemoryStream" nillable="true" type="q4:MemoryStream"/>
  <xs:complexType name="Stream">
    <xs:complexContent mixed="false">
      <xs:extension xmlns:q5="http://schemas.datacontract.org/2004/07/System"
base="q5:MarshalByRefObject">
        <xs:sequence/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="Stream" nillable="true" type="q4:Stream"/>
</xs:schema>
```

7.7 http://schemas.datacontract.org/2004/07/System Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:q5="http://schemas.datacontract.org/2004/07/System"
elementFormDefault="qualified"
targetNamespace="http://schemas.datacontract.org/2004/07/System"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="MarshalByRefObject">
    <xs:sequence>
      <xs:element name="__identity" nillable="true" type="xs:anyType"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="MarshalByRefObject" nillable="true" type="q5:MarshalByRefObject"/>
</xs:schema>
```

8 Appendix C: 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
- Microsoft SharePoint Server 2013
- Microsoft SharePoint Server 2016

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.

9 Change Tracking

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

10 Index