[MS-UPSGRAD]:
User Profile Service Push 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 iplg@microsoft.com.

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

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

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
04/04/2008	0.1		Initial Availability
06/27/2008	1.0	Major	Revised and edited the technical content
12/12/2008	1.01	Editorial	Revised and edited the technical content
07/13/2009	1.02	Major	Changes made for template compliance
08/28/2009	1.03	Editorial	Revised and edited the technical content
11/06/2009	1.04	Editorial	Revised and edited the technical content
02/19/2010	2.0	Editorial	Revised and edited the technical content
03/31/2010	2.01	Editorial	Revised and edited the technical content
04/30/2010	2.02	Editorial	Revised and edited the technical content
06/07/2010	2.03	Editorial	Revised and edited the technical content
06/29/2010	2.04	Editorial	Changed language and formatting in the technical content.
07/23/2010	2.05	Minor	Clarified the meaning of the technical content.
09/27/2010	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	2.05	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

Release: Sunday, December 19, 2010

1 Introduction

This document describes the User Profile Service Push protocol, which allows one user profile service to push updated user profiles and audiences to another user profile service.

1.1 Glossary

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

globally unique identifier (GUID)
security identifier (SID)
Security Support Provider Interface (SSPI)

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

audience
farm
login name
result set
return code
SQL (Structured Query Language)
stored procedure
T-SQL (Transact-Structured Query Language)
user profile
user profile change entry log

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. 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

[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

[MS-TDS] Microsoft Corporation, "Tabular Data Stream Protocol Specification", February 2008.

[MS-WSSFO2] Microsoft Corporation, "Windows SharePoint Services (WSS): File Operations Database Communications Version 2 Protocol Specification", July 2009.

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

Release: Sunday, December 19, 2010

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

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

[XMLNS] World Wide Web Consortium, "Namespaces in XML 1.0 (Third Edition)", W3C Recommendation 8 December 2009, http://www.w3.org/TR/REC-xml-names/

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

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

1.2.2   Informative References

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary", March 2007.

[MS-OFCGLOS] Microsoft Corporation, "Microsoft Office Master Glossary", June 2008.

[MS-UPSAUD] Microsoft Corporation, "User Profile Service Audiences Protocol Specification", June 2008.

[MS-UPSCHNG] Microsoft Corporation, "User Profile Change Log Stored Procedure Protocol Specification", June 2008.

1.3   Protocol Overview (Synopsis)

This protocol describes how the user profile service, acting as the protocol client, can communicate with a prior version of the user profile service, acting as the protocol server, to regularly update **user profiles** and **audience** details.

1.4   Relationship to Other Protocols

The following diagram shows the transport stack that the protocol uses:
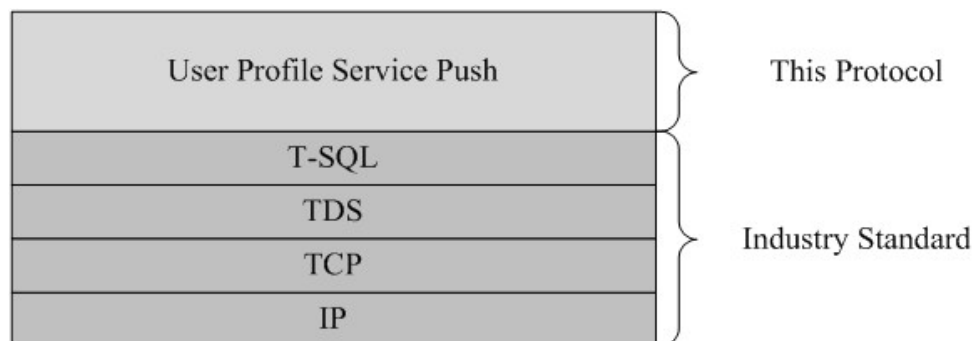

Figure 1: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

This protocol requires that a **farm** is installed and configured. The operations described by the protocol operate between a client and a backend database server on which the databases of the farm are stored.

The user that calls the **stored procedures** is required to have adequate permission to access the databases that contain the stored procedures.

1.6 Applicability Statement

This protocol is appropriate when there is simultaneously a prior version of the user profile service and a current version of the user profile service, and the prior version is considered to be read-only to its users. In this scenario, this protocol allows the current version to push updates to the prior version. This protocol is only appropriate as a temporary measure until all users of the prior version can be transitioned to the current version.

1.7 Versioning and Capability Negotiation

Versions of the data structures or stored procedures in the database are required to be the same as those expected by the front-end Web. 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.

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

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2   Messages

2.1   Transport

[MS-TDS] is the transport protocol used to call the stored procedures, to get **return codes**, and to return **result sets**.

2.2   Common Data Types

None.

2.2.1   Simple Data Types and Enumerations

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

2.2.2   Bit Fields and Flag Structures

No common bit field or flag structures are defined in this protocol.

2.2.3   Binary Structures

No common binary structures are defined in this protocol.

2.2.4   Result Sets

2.2.4.1   UpdateUserProfileBlobDataResult Result Set

The **profile_UpdateUserProfileBlobData** stored procedure returns an UpdateUserProfileBlobDataResult result set that contains information about the changes that were applied. The result set MUST contain 1 row. The UpdateUserProfileBlobDataResult result set is defined using T-SQL syntax, as follows:

```
UserID                  uniqueidentifier,
UpdateCount             int;
```

UserID: The identifier of the user profile that was created or updated.

UpdateCount: The number of user profile properties updated. MUST be 1.

2.2.4.2   UpdateUserProfileDataResult Result Set

The **profile_UpdateUserProfileData** stored procedure returns an UpdateUserProfileDataResult result set that contains details about the error information in case of failure or the count of properties updated in the case of a success. The UpdateUserProfileDataResult result set is defined using T-SQL syntax, as follows:

```
ERROR                   int,
XMLUpdateUserErr        int,
XMLUpdatePropertyErr    int,
UpdatePropertyCount     int,
NEWUSERGUID             GUID;
```

ERROR: Ignored. MUST always be 0.

XMLUpdateUserErr: Number of user profiles which failed to be updated. MUST be 0 or 1.

XMLUpdatePropertyErr: Number of user profile properties which failed to be updated. MUST be 0 or 1.

UpdatePropertyCount: Number of user profile properties which have been successfully updated. MUST be 0 or 1.

NEWUSERGUID: Ignored. MUST always be NULL.

2.2.5 Tables and Views

No common table or view structures are defined in this protocol.

2.2.6 XML Structures

No common XML structures are defined in this protocol.

3   Protocol Details

3.1   User Profile Service Push Protocol Server Details

3.1.1   Abstract Data Model

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

The protocol server stores information about user profiles and their values for each user profile property, and information about audiences. Each message processing event for this protocol is intended to update this set of information.

3.1.2   Timers

An execution timeout timer on the protocol server governs the execution time for the client's requests. The amount of time is specified by a timeout value that is configured on the protocol server for all connections.

3.1.3   Initialization

A connection that uses the underlying protocol layers that are specified in section 1.4: Relationship to Other Protocols MUST be established before using this protocol.

3.1.4   Higher-Layer Triggered Events

None.

3.1.5   Message Processing Events and Sequencing Rules

The following table summarizes the types that are defined in this specification.

Operation	Description
Update UserProfile UserIDs	Updates the identifier of a user profile.
Orgle_UpdateOrgleUpgrade	Updates the name, description, and memberships of an audience.
profile_UpdateUserProfileBlobData	Creates a new user profile.
profile_UpdateUserProfileData	Updates the user profile property values for a given user profile.

3.1.5.1   Update UserProfile UserIDs

The Update UserProfile UserIDs command is called to update the user profile identifier. Update UserProfile UserIDs is defined using **T-SQL** syntax, as follows:

```
exec sp_executesql N'update userprofile set UserID=@UserId where ntname=@AccountName',
N'@UserId uniqueidentifier,@AccountName nvarchar(400)', @UserId='<@UserId Value>',
@AccountName=N'<@AccountName Value>'
```

<@AccountName Value>: The **login name** that identifies the user profile to be updated. MUST be specified. MUST be a string with a maximum length of 400 characters.

<@UserId Value>: The updated identifier to apply to the user profile. MUST be a **GUID**. MUST NOT be NULL.

If there is a user profile with a login name matching the *@AccountName* parameter, then the user profile identifier MUST be set to the value of the *@UserId* parameter. If there is no matching user profile, then this command MUST NOT change any values.

Return Code Values: The Update UserProfile UserIDs command MUST return an integer return code with the value 0.

Result Sets: The Update UserProfile UserIDs command MUST NOT return a result set.

3.1.5.2 Orgle_UpdateOrgleUpgrade

The **Orgle_UpdateOrgleUpgrade** stored procedure is called to update the name, description, and members of a specified audience. **Orgle_UpdateOrgleUpgrade** is defined using T-SQL syntax, as follows:

```
PROCEDURE Orgle_UpdateOrgleUpgrade(
    @OrgleID                uniqueidentifier,
    @OrgleName              nvarchar(200),
    @Description            nvarchar(1500),
    @ItemsXml               ntext
);
```

@OrgleID: The identifier of the audience to be updated. MUST NOT be NULL.

@OrgleName: The name of the audience. MUST be specified and MUST be less than 200 characters.

@Description: The description of the audience. MUST be less than 1500 characters.

@ItemsXml: The list of members of the audience. MUST be specified. MUST conform to the XML schema as follows ([XML10], [XMLNS], [XMLINFOSET], [XMLSCHEMA1], [XMLSCHEMA2]).

```
<?xml version="1.0" encoding="utf-16"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Orgle">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Result" minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
            <xs:attribute name="NTName" type="xs:string"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Result: The set of members of the audience.

NTName: The login name of a user who is a member of the specified audience. MUST not be NULL. MUST be less than 400 characters.

If there is an audience with an identifier matching the *@OrgleID* parameter, then that audience name, description, and list of members MUST be updated based on the parameters. This stored procedure MUST remove all existing members of the audience, and replace them with the list of users specified in the *@ItemsXml* parameter. If there is no matching audience, then this stored procedure MUST NOT change any values.

Return Code Values: Orgle_UpdateOrgleUpgrade MUST return an integer return code with the value 0.

Result Sets: Orgle_UpdateOrgleUpgrade MUST NOT return a result set.

3.1.5.3   profile_UpdateUserProfileBlobData

The **profile_UpdateUserProfileBlobData** stored procedure is called to create a new user profile. **profile_UpdateUserProfileBlobData** is defined using T-SQL syntax, as follows:

```
PROCEDURE profile_UpdateUserProfileBlobData (
    @UserID                  uniqueidentifier,
    @NTName                  nvarchar(400),
    @PropertyName            nvarchar(250),
    @PropertyValVarbinary    varbinary(7500) = null,
    @PropertyValText         ntext = null,
    @PropertyValImage        image = null,
    @Debug                   bit = 0
);
```

@UserID: Reserved. MUST be NULL.

@NTName: The login name for the user whose user profile is to be created. MUST NOT be NULL. MUST be less than 400 characters.

@PropertyName: Reserved. MUST be the string "SID".

@PropertyValVarbinary: The binary data value of the **security identifier (SID)** of the user. MUST be specified. MUST NOT be NULL.

@PropertyValText: This parameter MUST NOT be specified and MUST be ignored by the server.

@PropertyValImage: This parameter MUST NOT be specified and MUST be ignored by the server.

@Debug: This parameter MUST NOT be specified and MUST be ignored by the server.

If there is no existing user profile with a security identifier matching the *@PropertyValVarbinary* parameter, then the user profile will be created. If there is already a matching user profile, then this stored procedure MUST NOT change any values.

Return Code Values: profile_UpdateUserProfileBlobData MUST return an integer return code with the value 0.

Result Sets: profile_UpdateUserProfileBlobData MUST return the UpdateUserProfileBlobDataResult result set.

3.1.5.4   profile_UpdateUserProfileData

The **profile_UpdateUserProfileData** stored procedure is called to update a single value for a single user profile property corresponding to the specified user profile. **profile_UpdateUserProfileData** is defined using T-SQL syntax, as follows:

```
PROCEDURE profile_UpdateUserProfileData(
    @UpdatePropertyList              ntext,
    @Debug                           bit = 0,
);
```

@UpdatePropertyList: Contains the user profile property and the new value that is to be applied. This parameter MUST be specified and MUST adhere to the following XML schema ([XML10], [XMLNS], [XMLINFOSET], [XMLSCHEMA1], [XMLSCHEMA2]).

```
<?xml version="1.0" encoding="utf-16"?>
<xs:schema id="MSPROFILE" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="MSPROFILE">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="PROFILE">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="USER">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="PROPERTY">
                      <xs:complexType>
                        <xs:attribute name="PropertyName"
                                      type="xs:string" use="required" />
                        <xs:attribute name="PropertyValue" type="xs:string" />
                        <xs:attribute name="RemoveFlag" type="xs:int" />
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                  <xs:attribute name="UserID" type="xs:string" use="required" />
                  <xs:attribute name="NTAccount" type="xs:string" use="required" />
                  <xs:attribute name="NewUser" type="xs:int" use="required" />
                </xs:complexType>
              </xs:element>
            </xs:sequence>
            <xs:attribute name="ProfileName" type="xs:string" use="required"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

MSPROFILE.PROFILE.USER.UserID: The identifier of the user profile to be updated. MUST be a GUID.

MSPROFILE.PROFILE.USER.NTAccount: The login name of the user profile to be updated. MUST NOT be NULL and MUST be less than 400 characters.

MSPROFILE.PROFILE.USER.NewUser: This parameter MUST be 0 and MUST be ignored by the server.

MSPROFILE.PROFILE.USER.PROPERTY.PropertyName: The name of the user profile property to be updated. MUST NOT be NULL and MUST be less than 250 characters.

MSPROFILE.PROFILE.USER.PROPERTY.PropertyValue: The new value of the user profile property to be applied to the specified user profile. MUST be less than 3600 characters.

MSPROFILE.PROFILE.USER.PROPERTY.RemoveFlag: Whether the existing value for the specified user profile property should be removed. The value MUST NOT be specified when applying the value specified in PropertyValue, and MUST be 1 to remove any existing value. When RemoveFlag is 1, PropertyValue MUST be NULL.

@Debug: This parameter MUST NOT be specified and MUST be ignored by the server.

If there is a user profile with an identifier matching the **MSPROFILE.PROFILE.USER.UserID** attribute value, then that user profile MUST be updated. If there is no matching user profile, then this stored procedure MUST NOT change any values.

Return Code Values: profile_UpdateUserProfileData MUST return an integer return code with the value 0.

Result Sets: profile_UpdateUserProfileData MUST return the UpdateUserProfileDataResult result set.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 User Profile Service Push Client Details

3.2.1 Abstract Data Model

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

The protocol client stores information about user profiles and their values for each user profile property, and information about audiences.

The protocol client also maintains a **user profile change entry log** to keep track of changes to user profiles.

3.2.2 Timers

The protocol client MUST create a timer to regularly update the protocol server. The timer MUST run once per hour.

3.2.3   Initialization

The protocol client MUST be initialized with the list of the user profile properties to be updated on the protocol server, such that calls to **profile_UpdateUserProfileData** will only be made for user profile properties that actually exist.

3.2.4   Higher-Layer Triggered Events

None.

3.2.5   Message Processing Events and Sequencing Rules

None.

3.2.6   Timer Events

The timer created by the protocol client, as described in section 3.2.2, MUST discover when user profiles are created or updated on the protocol client, and create new user profiles and update user profile property values on the protocol server. The timer MUST load all audiences on the protocol client and update audience information about the protocol server.

3.2.7   Other Local Events

None.

4   Protocol Examples

4.1   User Profiles

User profiles can be kept up to date by using the [MS-UPSCHNG] protocol to detect the latest changes to any user profiles.

As an example, consider the case where a new user is detected with login name "contoso\john", UserID "07010AA8-9347-4D6D-A9B9-4C61EB57A9E3**"** and SID "0x01050000000000515000000A065CF7E784B9B5FE77C877003D32000".

When the new user profile is detected, the protocol client calls **profile_UpdateUserProfileBlobData** with the user profile login name and the SID to create the user profile on the protocol server:

```
exec dbo.profilesynch_UpdateUserProfileBlobData
    @UserID = null,
    @NTName = 'contoso\john',
    @PropertyName = 'SID',
      @PropertyValVarbinary =
          '0x01050000000000515000000A065CF7E784B9B5FE77C877003D32000'
```

The protocol client then calls the **Update UserProfile UserIDs** command to update the user's ID on the protocol server:

```
exec sp_executesql
    N'update userprofile set UserID=@UserId where ntname=@AccountName',
    N'@UserId uniqueidentifier,@AccountName nvarchar(400)',
    @UserId='07010AA8-9347-4D6D-A9B9-4C61EB57A9E3',
    @AccountName=N'contoso\john'
```

Now suppose the user's "FirstName" user profile property is changed to "John".

When the change to a user profile property is detected, the protocol client calls **profile_UpdateUserProfileData** with the user profile identifier to update the user profile property value on the protocol server:

```
exec dbo.profilesynch_UpdateUserProfileData
    @UpdatePropertyList = '
        <?xml version="1.0" encoding="utf-16"?>
        <MSPROFILE>
          <PROFILE ProfileName="UserProfile">
            <USER NewUser="0" NTAccount="contoso\john"
                  UserID="07010AA8-9347-4D6D-A9B9-4C61EB57A9E3">
                <PROPERTY PropertyName="FirstName" PropertyValue="John" />
            </USER>
          </PROFILE>
        </MSPROFILE>'
```

4.2 Audiences

Audiences can be kept up to date by using the [MS-UPSAUD] protocol to load the latest details for each audience.

As an example, consider the case where a new audience is detected with name "John's reports", description "People who report to John", ID "69D93917-9BAD-4F8E-9A78-98685246ADB6", and members "contoso\lori" and "contoso\sara".

When the new audience is detected, the protocol client calls **orgle_UpdateOrgleUpgrade** for the audience to update the name, description, and memberships on the protocol server:

```
exec dbo.Orgle_UpdateOrgleUpgrade
    @OrgleID =  '69D93917-9BAD-4F8E-9A78-98685246ADB6',
    @OrgleName = 'John's reports'
    @Description = 'People who report to John',
    @ItemsXml = '
        <?xml version="1.0" encoding="utf-16"?>
        <Orgle>
            <Result NTName = "contoso\lori" />
            <Result NTName = "contoso\sara" />
        </Orgle>'
```

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 prior to invoking the stored procedure

5.2   Index of Security Parameters

None.

6   Appendix A: Product Behavior

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

- Microsoft® SharePoint® 2010 for Internet Sites Enterprise

- Microsoft® SharePoint® 2010 for Internet Sites Standard

- Microsoft® Office SharePoint® Server 2007

- 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