

# [MS-USPSWS]:

## User Profile Service 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](mailto:iplg@microsoft.com).
- **License Programs.** To see all of the protocols in scope under a specific license program and the associated patents, visit the [Patent Map](#).
- **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](http://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.

**Support.** For questions and support, please contact [dochelp@microsoft.com](mailto:dochelp@microsoft.com).

## Revision Summary

Date	Revision History	Revision Class	Comments
4/4/2008	0.1	New	Initial Availability
6/27/2008	1.0	Major	Revised and edited the technical content
8/15/2008	1.01	Editorial	Revised and edited the technical content
12/12/2008	1.02	Editorial	Revised and edited the technical content
7/13/2009	1.03	Major	Revised and edited the technical content
8/28/2009	1.04	Editorial	Revised and edited the technical content
11/6/2009	1.05	Editorial	Revised and edited the technical content
2/19/2010	2.0	Major	Updated and revised the technical content
3/31/2010	2.01	Editorial	Revised and edited the technical content
4/30/2010	2.02	Editorial	Revised and edited the technical content
6/7/2010	2.03	Editorial	Revised and edited the technical content
6/29/2010	2.04	Editorial	Changed language and formatting in the technical content.
7/23/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
9/27/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
3/18/2011	2.04	None	No changes to the meaning, language, or formatting of the technical content.
6/10/2011	2.04	None	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	3.0	Major	Significantly changed the technical content.
4/11/2012	3.0	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	3.1	Minor	Clarified the meaning of the technical content.
9/12/2012	3.1	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	4.0	Major	Significantly changed the technical content.
2/11/2013	4.0	None	No changes to the meaning, language, or formatting of the technical content.
7/30/2013	4.1	Minor	Clarified the meaning of the technical content.
11/18/2013	4.1	None	No changes to the meaning, language, or formatting of the

<b>Date</b>	<b>Revision History</b>	<b>Revision Class</b>	<b>Comments</b>
			technical content.
2/10/2014	4.1	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	4.2	Minor	Clarified the meaning of the technical content.
7/31/2014	4.2	None	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	4.2	None	No changes to the meaning, language, or formatting of the technical content.
3/16/2015	5.0	Major	Significantly changed the technical content.
2/26/2016	6.0	Major	Significantly changed the technical content.
7/15/2016	6.0	None	No changes to the meaning, language, or formatting of the technical content.
9/14/2016	6.0	None	No changes to the meaning, language, or formatting of the technical content.
6/20/2017	7.0	Major	Significantly changed the technical content.
7/24/2018	8.0	Major	Significantly changed the technical content.
10/1/2018	9.0	Major	Significantly changed the technical content.
6/18/2019	9.0	None	No changes to the meaning, language, or formatting of the technical content.
2/16/2021	9.1	Minor	Clarified the meaning of the technical content.
4/22/2021	10.0	Major	Significantly changed the technical content.
7/20/2021	11.0	Major	Significantly changed the technical content.
8/17/2021	12.0	Major	Significantly changed the technical content.

# Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>11</b>
1.1	Glossary .....	11
1.2	References .....	13
1.2.1	Normative References .....	13
1.2.2	Informative References .....	14
1.3	Protocol Overview (Synopsis) .....	14
1.4	Relationship to Other Protocols .....	14
1.5	Prerequisites/Preconditions .....	15
1.6	Applicability Statement .....	15
1.7	Versioning and Capability Negotiation .....	15
1.8	Vendor-Extensible Fields .....	15
1.9	Standards Assignments.....	15
<b>2</b>	<b>Messages.....</b>	<b>16</b>
2.1	Transport .....	16
2.2	Common Message Syntax .....	16
2.2.1	Namespaces .....	16
2.2.2	Messages.....	16
2.2.3	Elements .....	16
2.2.4	Complex Types.....	16
2.2.4.1	ArrayOfContactData .....	17
2.2.4.2	ArrayOfDouble .....	17
2.2.4.3	ArrayOfMembershipData .....	18
2.2.4.4	ArrayOfOrganizationProfileData.....	18
2.2.4.5	ArrayOfPinnedLinkData .....	18
2.2.4.6	ArrayOfPropertyData .....	18
2.2.4.7	ArrayOfPropertyInfo .....	19
2.2.4.8	ArrayOfQuickLinkData .....	19
2.2.4.9	ArrayOfString .....	19
2.2.4.10	ArrayOfValueData .....	19
2.2.4.11	ContactData .....	20
2.2.4.12	MemberGroupData .....	20
2.2.4.13	MembershipData.....	21
2.2.4.14	OrganizationProfileData .....	22
2.2.4.15	PinnedLinkData.....	22
2.2.4.16	PropertyData .....	22
2.2.4.17	PropertyInfo .....	23
2.2.4.18	QuickLinkData .....	24
2.2.4.19	ValueData .....	25
2.2.5	Simple Types .....	25
2.2.5.1	ChoiceTypes .....	25
2.2.5.2	guid .....	26
2.2.5.3	MembershipSource.....	26
2.2.5.4	Privacy .....	27
2.2.5.5	SuggestionType .....	27
2.2.6	Attributes .....	28
2.2.7	Groups .....	28
2.2.8	Attribute Groups.....	28
<b>3</b>	<b>Protocol Details .....</b>	<b>29</b>
3.1	Protocol Server Details .....	29
3.1.1	Abstract Data Model.....	29
3.1.2	Timers .....	32
3.1.3	Initialization.....	32
3.1.4	Message Processing Events and Sequencing Rules .....	33

3.1.4.1	AddColleague .....	34
3.1.4.1.1	Messages .....	35
3.1.4.1.1.1	AddColleagueSoapIn .....	35
3.1.4.1.1.2	AddColleagueSoapOut .....	35
3.1.4.1.2	Elements .....	35
3.1.4.1.2.1	AddColleague .....	35
3.1.4.1.2.2	AddColleagueResponse .....	36
3.1.4.2	AddColleagueWithoutEmailNotification .....	37
3.1.4.2.1	Messages .....	37
3.1.4.2.1.1	AddColleagueWithoutEmailNotificationSoapIn .....	37
3.1.4.2.1.2	AddColleagueWithoutEmailNotificationSoapOut .....	37
3.1.4.2.2	Elements .....	38
3.1.4.2.2.1	AddColleagueWithoutEmailNotification .....	38
3.1.4.2.2.2	AddColleagueWithoutEmailNotificationResponse .....	39
3.1.4.3	AddLink .....	39
3.1.4.3.1	Messages .....	39
3.1.4.3.1.1	AddLinkSoapIn .....	39
3.1.4.3.1.2	AddLinkSoapOut .....	39
3.1.4.3.2	Elements .....	40
3.1.4.3.2.1	AddLink .....	40
3.1.4.3.2.2	AddLinkResponse .....	40
3.1.4.4	AddMembership .....	41
3.1.4.4.1	Messages .....	41
3.1.4.4.1.1	AddMembershipSoapIn .....	41
3.1.4.4.1.2	AddMembershipSoapOut .....	41
3.1.4.4.2	Elements .....	41
3.1.4.4.2.1	AddMembership .....	41
3.1.4.4.2.2	AddMembershipResponse .....	42
3.1.4.5	AddPinnedLink .....	42
3.1.4.5.1	Messages .....	43
3.1.4.5.1.1	AddPinnedLinkSoapIn .....	43
3.1.4.5.1.2	AddPinnedLinkSoapOut .....	43
3.1.4.5.2	Elements .....	43
3.1.4.5.2.1	AddPinnedLink .....	43
3.1.4.5.2.2	AddPinnedLinkResponse .....	44
3.1.4.6	AddSuggestions .....	44
3.1.4.6.1	Messages .....	44
3.1.4.6.1.1	AddSuggestionsSoapIn .....	44
3.1.4.6.1.2	AddSuggestionsSoapOut .....	45
3.1.4.6.2	Elements .....	45
3.1.4.6.2.1	AddSuggestions .....	45
3.1.4.6.2.2	AddSuggestionsResponse .....	45
3.1.4.7	CreateMemberGroup .....	46
3.1.4.7.1	Messages .....	46
3.1.4.7.1.1	CreateMemberGroupSoapIn .....	46
3.1.4.7.1.2	CreateMemberGroupSoapOut .....	46
3.1.4.7.2	Elements .....	46
3.1.4.7.2.1	CreateMemberGroup .....	46
3.1.4.7.2.2	CreateMemberGroupResponse .....	47
3.1.4.8	CreateUserProfileByAccountName .....	47
3.1.4.8.1	Messages .....	47
3.1.4.8.1.1	CreateUserProfileByAccountNameSoapIn .....	47
3.1.4.8.1.2	CreateUserProfileByAccountNameSoapOut .....	48
3.1.4.8.2	Elements .....	48
3.1.4.8.2.1	CreateUserProfileByAccountName .....	48
3.1.4.8.2.2	CreateUserProfileByAccountNameResponse .....	48
3.1.4.9	GetCommonColleagues .....	49
3.1.4.9.1	Messages .....	49

3.1.4.9.1.1	GetCommonColleaguesSoapIn .....	49
3.1.4.9.1.2	GetCommonColleaguesSoapOut .....	49
3.1.4.9.2	Elements .....	49
3.1.4.9.2.1	GetCommonColleagues .....	49
3.1.4.9.2.2	GetCommonColleaguesResponse .....	50
3.1.4.10	GetCommonManager .....	50
3.1.4.10.1	Messages .....	51
3.1.4.10.1.1	GetCommonManagerSoapIn .....	51
3.1.4.10.1.2	GetCommonManagerSoapOut .....	51
3.1.4.10.2	Elements .....	51
3.1.4.10.2.1	GetCommonManager .....	51
3.1.4.10.2.2	GetCommonManagerResponse .....	52
3.1.4.11	GetCommonMemberships .....	52
3.1.4.11.1	Messages .....	52
3.1.4.11.1.1	GetCommonMembershipsSoapIn .....	52
3.1.4.11.1.2	GetCommonMembershipsSoapOut .....	53
3.1.4.11.2	Elements .....	53
3.1.4.11.2.1	GetCommonMemberships .....	53
3.1.4.11.2.2	GetCommonMembershipsResponse .....	53
3.1.4.12	GetInCommon .....	54
3.1.4.12.1	Messages .....	54
3.1.4.12.1.1	GetInCommonSoapIn .....	54
3.1.4.12.1.2	GetInCommonSoapOut .....	54
3.1.4.12.2	Elements .....	54
3.1.4.12.2.1	GetInCommon .....	54
3.1.4.12.2.2	GetInCommonResponse .....	55
3.1.4.12.3	Complex Types .....	55
3.1.4.12.3.1	InCommonData .....	55
3.1.4.13	GetProfileSchema .....	56
3.1.4.13.1	Messages .....	56
3.1.4.13.1.1	GetProfileSchemaSoapIn .....	56
3.1.4.13.1.2	GetProfileSchemaSoapOut .....	56
3.1.4.13.2	Elements .....	56
3.1.4.13.2.1	GetProfileSchema .....	57
3.1.4.13.2.2	GetProfileSchemaResponse .....	57
3.1.4.14	GetProfileSchemaNameByAccountName .....	57
3.1.4.14.1	Messages .....	58
3.1.4.14.1.1	GetProfileSchemaNameByAccountNameSoapIn .....	58
3.1.4.14.1.2	GetProfileSchemaNameByAccountNameSoapOut .....	58
3.1.4.14.2	Elements .....	58
3.1.4.14.2.1	GetProfileSchemaNameByAccountName .....	58
3.1.4.14.2.2	GetProfileSchemaNameByAccountNameResponse .....	59
3.1.4.15	GetProfileSchemaNames .....	59
3.1.4.15.1	Messages .....	59
3.1.4.15.1.1	GetProfileSchemaNamesSoapIn .....	59
3.1.4.15.1.2	GetProfileSchemaNamesSoapOut .....	60
3.1.4.15.2	Elements .....	60
3.1.4.15.2.1	GetProfileSchemaNames .....	60
3.1.4.15.2.2	GetProfileSchemaNamesResponse .....	60
3.1.4.16	GetPropertyChoiceList .....	60
3.1.4.16.1	Messages .....	61
3.1.4.16.1.1	GetPropertyChoiceListSoapIn .....	61
3.1.4.16.1.2	GetPropertyChoiceListSoapOut .....	61
3.1.4.16.2	Elements .....	61
3.1.4.16.2.1	GetPropertyChoiceList .....	61
3.1.4.16.2.2	GetPropertyChoiceListResponse .....	62
3.1.4.17	GetUserColleagues .....	62
3.1.4.17.1	Messages .....	62

3.1.4.17.1.1	GetUserColleaguesSoapIn .....	62
3.1.4.17.1.2	GetUserColleaguesSoapOut .....	63
3.1.4.17.2	Elements .....	63
3.1.4.17.2.1	GetUserColleagues .....	63
3.1.4.17.2.2	GetUserColleaguesResponse .....	63
3.1.4.18	GetUserLinks .....	64
3.1.4.18.1	Messages .....	64
3.1.4.18.1.1	GetUserLinksSoapIn .....	64
3.1.4.18.1.2	GetUserLinksSoapOut .....	64
3.1.4.18.2	Elements .....	64
3.1.4.18.2.1	GetUserLinks .....	64
3.1.4.18.2.2	GetUserLinksResponse .....	65
3.1.4.19	GetUserMemberships .....	65
3.1.4.19.1	Messages .....	66
3.1.4.19.1.1	GetUserMembershipsSoapIn .....	66
3.1.4.19.1.2	GetUserMembershipsSoapOut .....	66
3.1.4.19.2	Elements .....	66
3.1.4.19.2.1	GetUserMemberships .....	66
3.1.4.19.2.2	GetUserMembershipsResponse .....	67
3.1.4.20	GetUserOrganizations .....	67
3.1.4.20.1	Messages .....	68
3.1.4.20.1.1	GetUserOrganizationsSoapIn .....	68
3.1.4.20.1.2	GetUserOrganizationsSoapOut .....	68
3.1.4.20.2	Elements .....	68
3.1.4.20.2.1	GetUserOrganizations .....	68
3.1.4.20.2.2	GetUserOrganizationsResponse .....	69
3.1.4.21	GetUserPinnedLinks .....	69
3.1.4.21.1	Messages .....	69
3.1.4.21.1.1	GetUserPinnedLinksSoapIn .....	69
3.1.4.21.1.2	GetUserPinnedLinksSoapOut .....	70
3.1.4.21.2	Elements .....	70
3.1.4.21.2.1	GetUserPinnedLinks .....	70
3.1.4.21.2.2	GetUserPinnedLinksResponse .....	70
3.1.4.22	GetUserProfileByGuid .....	71
3.1.4.22.1	Messages .....	71
3.1.4.22.1.1	GetUserProfileByGuidSoapIn .....	71
3.1.4.22.1.2	GetUserProfileByGuidSoapOut .....	71
3.1.4.22.2	Elements .....	72
3.1.4.22.2.1	GetUserProfileByGuid .....	72
3.1.4.22.2.2	GetUserProfileByGuidResponse .....	72
3.1.4.23	GetUserProfileByIndex .....	72
3.1.4.23.1	Messages .....	73
3.1.4.23.1.1	GetUserProfileByIndexSoapIn .....	73
3.1.4.23.1.2	GetUserProfileByIndexSoapOut .....	73
3.1.4.23.2	Elements .....	73
3.1.4.23.2.1	GetUserProfileByIndex .....	73
3.1.4.23.2.2	GetUserProfileByIndexResponse .....	74
3.1.4.23.3	Complex Types .....	74
3.1.4.23.3.1	GetUserProfileByIndexResult .....	74
3.1.4.24	GetUserProfileByName .....	75
3.1.4.24.1	Messages .....	75
3.1.4.24.1.1	GetUserProfileByNameSoapIn .....	75
3.1.4.24.1.2	GetUserProfileByNameSoapOut .....	76
3.1.4.24.2	Elements .....	76
3.1.4.24.2.1	GetUserProfileByName .....	76
3.1.4.24.2.2	GetUserProfileByNameResponse .....	76
3.1.4.25	GetUserProfileCount .....	77
3.1.4.25.1	Messages .....	77

3.1.4.25.1.1	GetUserProfileCountSoapIn .....	77
3.1.4.25.1.2	GetUserProfileCountSoapOut .....	77
3.1.4.25.2	Elements .....	77
3.1.4.25.2.1	GetUserProfileCount .....	77
3.1.4.25.2.2	GetUserProfileCountResponse .....	78
3.1.4.26	GetUserProfileSchema .....	78
3.1.4.26.1	Messages .....	78
3.1.4.26.1.1	GetUserProfileSchemaSoapIn .....	78
3.1.4.26.1.2	GetUserProfileSchemaSoapOut .....	79
3.1.4.26.2	Elements .....	79
3.1.4.26.2.1	GetUserProfileSchema .....	79
3.1.4.26.2.2	GetUserProfileSchemaResponse .....	79
3.1.4.27	GetUserPropertyByAccountName .....	79
3.1.4.27.1	Messages .....	80
3.1.4.27.1.1	GetUserPropertyByAccountNameSoapIn .....	80
3.1.4.27.1.2	GetUserPropertyByAccountNameSoapOut .....	80
3.1.4.27.2	Elements .....	80
3.1.4.27.2.1	GetUserPropertyByAccountName .....	80
3.1.4.27.2.2	GetUserPropertyByAccountNameResponse .....	81
3.1.4.28	ModifyUserPropertyByAccountName .....	81
3.1.4.28.1	Messages .....	82
3.1.4.28.1.1	ModifyUserPropertyByAccountNameSoapIn .....	82
3.1.4.28.1.2	ModifyUserPropertyByAccountNameSoapOut .....	82
3.1.4.28.2	Elements .....	82
3.1.4.28.2.1	ModifyUserPropertyByAccountName .....	82
3.1.4.28.2.2	ModifyUserPropertyByAccountNameResponse .....	83
3.1.4.29	RemoveAllColleagues.....	83
3.1.4.29.1	Messages .....	83
3.1.4.29.1.1	RemoveAllColleaguesSoapIn.....	83
3.1.4.29.1.2	RemoveAllColleaguesSoapOut .....	84
3.1.4.29.2	Elements .....	84
3.1.4.29.2.1	RemoveAllColleagues .....	84
3.1.4.29.2.2	RemoveAllColleaguesResponse .....	84
3.1.4.30	RemoveAllLinks .....	84
3.1.4.30.1	Messages .....	85
3.1.4.30.1.1	RemoveAllLinksSoapIn .....	85
3.1.4.30.1.2	RemoveAllLinksSoapOut .....	85
3.1.4.30.2	Elements .....	85
3.1.4.30.2.1	RemoveAllLinks .....	85
3.1.4.30.2.2	RemoveAllLinksResponse .....	86
3.1.4.31	RemoveAllMemberships .....	86
3.1.4.31.1	Messages .....	86
3.1.4.31.1.1	RemoveAllMembershipsSoapIn .....	86
3.1.4.31.1.2	RemoveAllMembershipsSoapOut .....	87
3.1.4.31.2	Elements .....	87
3.1.4.31.2.1	RemoveAllMemberships .....	87
3.1.4.31.2.2	RemoveAllMembershipsResponse .....	87
3.1.4.32	RemoveAllPinnedLinks .....	87
3.1.4.32.1	Messages .....	88
3.1.4.32.1.1	RemoveAllPinnedLinksSoapIn .....	88
3.1.4.32.1.2	RemoveAllPinnedLinksSoapOut .....	88
3.1.4.32.2	Elements .....	88
3.1.4.32.2.1	RemoveAllPinnedLinks .....	88
3.1.4.32.2.2	RemoveAllPinnedLinksResponse.....	89
3.1.4.33	RemoveColleague .....	89
3.1.4.33.1	Messages .....	89
3.1.4.33.1.1	RemoveColleagueSoapIn .....	89
3.1.4.33.1.2	RemoveColleagueSoapOut .....	90



3.1.4.33.2	Elements .....	90
3.1.4.33.2.1	RemoveColleague .....	90
3.1.4.33.2.2	RemoveColleagueResponse .....	90
3.1.4.34	RemoveLink .....	90
3.1.4.34.1	Messages .....	91
3.1.4.34.1.1	RemoveLinkSoapIn .....	91
3.1.4.34.1.2	RemoveLinkSoapOut .....	91
3.1.4.34.2	Elements .....	91
3.1.4.34.2.1	RemoveLink .....	91
3.1.4.34.2.2	RemoveLinkResponse .....	92
3.1.4.35	RemoveMembership .....	92
3.1.4.35.1	Messages .....	92
3.1.4.35.1.1	RemoveMembershipSoapIn .....	92
3.1.4.35.1.2	RemoveMembershipSoapOut .....	93
3.1.4.35.2	Elements .....	93
3.1.4.35.2.1	RemoveMembership .....	93
3.1.4.35.2.2	RemoveMembershipResponse .....	93
3.1.4.36	RemovePinnedLink .....	94
3.1.4.36.1	Messages .....	94
3.1.4.36.1.1	RemovePinnedLinkSoapIn .....	94
3.1.4.36.1.2	RemovePinnedLinkSoapOut .....	94
3.1.4.36.2	Elements .....	94
3.1.4.36.2.1	RemovePinnedLink .....	94
3.1.4.36.2.2	RemovePinnedLinkResponse .....	95
3.1.4.37	UpdateColleaguePrivacy .....	95
3.1.4.37.1	Messages .....	96
3.1.4.37.1.1	UpdateColleaguePrivacySoapIn .....	96
3.1.4.37.1.2	UpdateColleaguePrivacySoapOut .....	96
3.1.4.37.2	Elements .....	96
3.1.4.37.2.1	UpdateColleaguePrivacy .....	96
3.1.4.37.2.2	UpdateColleaguePrivacyResponse .....	97
3.1.4.38	UpdateLink .....	97
3.1.4.38.1	Messages .....	98
3.1.4.38.1.1	UpdateLinkSoapIn .....	98
3.1.4.38.1.2	UpdateLinkSoapOut .....	98
3.1.4.38.2	Elements .....	98
3.1.4.38.2.1	UpdateLink .....	98
3.1.4.38.2.2	UpdateLinkResponse .....	99
3.1.4.39	UpdateMembershipPrivacy .....	99
3.1.4.39.1	Messages .....	99
3.1.4.39.1.1	UpdateMembershipPrivacySoapIn .....	99
3.1.4.39.1.2	UpdateMembershipPrivacySoapOut .....	100
3.1.4.39.2	Elements .....	100
3.1.4.39.2.1	UpdateMembershipPrivacy .....	100
3.1.4.39.2.2	UpdateMembershipPrivacyResponse .....	101
3.1.4.40	UpdatePinnedLink .....	101
3.1.4.40.1	Messages .....	101
3.1.4.40.1.1	UpdatePinnedLinkSoapIn .....	101
3.1.4.40.1.2	UpdatePinnedLinkSoapOut .....	102
3.1.4.40.2	Elements .....	102
3.1.4.40.2.1	UpdatePinnedLink .....	102
3.1.4.40.2.2	UpdatePinnedLinkResponse .....	102
3.1.5	Timer Events .....	103
3.1.6	Other Local Events .....	103
<b>4</b>	<b>Protocol Examples .....</b>	<b>104</b>
4.1	Create User Profile Example .....	104
4.1.1	Create the Profile .....	104

4.1.1.1	Request .....	104
4.1.1.2	Response .....	104
4.2	Commonalities Example .....	105
4.2.1	GetCommonManager.....	105
4.2.1.1	Request .....	105
4.2.1.2	Response .....	105
4.2.2	GetCommonMemberships .....	106
4.2.2.1	Request .....	106
4.2.2.2	Response .....	106
4.2.3	GetCommonColleagues.....	107
4.2.3.1	Request .....	107
4.2.3.2	Response .....	107
4.2.4	GetInCommon.....	108
4.2.4.1	Request .....	108
4.2.4.2	Response .....	108
4.3	GetUserProfile Example.....	109
4.3.1	Retrieve Profile Definition .....	110
4.3.1.1	Obtain the Schema.....	110
4.3.1.1.1	Request .....	110
4.3.1.1.2	Response .....	110
4.3.1.2	Obtain the Profile of the Specified User .....	111
4.3.1.2.1	Request .....	111
4.3.1.2.2	Response .....	111
4.4	GetUserProfileByIndex Example .....	112
4.4.1	Enumerate Users .....	112
4.4.1.1	Obtain User Profiles.....	112
4.4.1.1.1	Request .....	112
4.4.1.1.2	Response .....	113
4.5	Memberships .....	116
4.5.1	Example Data .....	116
4.5.2	Synchronize Two Users.....	117
4.5.2.1	Obtain Memberships.....	117
4.5.2.1.1	Requests .....	117
4.5.2.1.2	Responses .....	118
4.5.2.2	Adding Memberships to P1 – Martin Weber.....	119
4.5.2.2.1	Requests .....	119
4.5.2.2.2	Responses .....	120
4.5.2.3	Deleting Memberships from P1 – Martin Weber .....	120
4.5.2.3.1	Requests .....	121
4.5.2.3.2	Responses .....	121
<b>5</b>	<b>Security .....</b>	<b>122</b>
5.1	Security Considerations for Implementers .....	122
5.2	Index of Security Parameters .....	122
<b>6</b>	<b>Appendix A: Full WSDL .....</b>	<b>123</b>
<b>7</b>	<b>Appendix B: Product Behavior .....</b>	<b>160</b>
<b>8</b>	<b>Change Tracking.....</b>	<b>162</b>
<b>9</b>	<b>Index.....</b>	<b>163</b>

# 1 Introduction

The User Profile Service Web Service Protocol enables a protocol client to retrieve and manage properties that pertain to a specific person or entity within a portal site, and also data in common among users. This protocol also enables a protocol client to manage links, group memberships, and users who have a social networking relationship with another user.

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:

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

**Colleague Tracker Web Part:** A type of Web Part that users can add to a My Site to track changes to the profiles of their colleagues.

**contact:** (1) A presence entity (presentity) whose presence information can be tracked.

(2) A person, company, or other entity that is stored in a directory and is associated with one or more unique identifiers and attributes, such as an Internet message address or login name.

**current user:** The user who is authenticated during processing operations on a front-end web server or a back-end database server.

**directory service (DS):** A service that stores and organizes information about a computer network's users and network shares, and that allows network administrators to manage users' access to the shares. See also Active Directory.

**distribution list:** A collection of users, computers, contacts, or other groups that is used only for email distribution, and addressed as a single recipient.

**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).

**group:** (1) A named collection of users who share similar access permissions or roles.

(2) A named collection of quick links, colleagues, or memberships for the purpose of organization.

**Hypertext Transfer Protocol (HTTP):** An application-level protocol for distributed, collaborative, hypermedia information systems (text, graphic images, sound, video, and other multimedia files) on the World Wide Web.

**Hypertext Transfer Protocol Secure (HTTPS):** An extension of HTTP that securely encrypts and decrypts web page requests. In some older protocols, "Hypertext Transfer Protocol over Secure Sockets Layer" is still used (Secure Sockets Layer has been deprecated). For more information, see [\[SSL3\]](#) and [\[RFC5246\]](#).

**keyword:** One or more words or phrases that site administrators identified as important for a search service application. A keyword provides a way to apply business rules to search results for queries that use the keyword.

**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".

**member group:** A group of users that is specific to a user profile service. Examples of types of member groups are distribution lists, security groups, and SharePoint sites. A member group contains metadata such as the group name, email address, URL, and a list of members.

**membership:** The state or status of being a member of a member group. A membership contains additional metadata such as the privacy level that is associated with the membership.

**organization:** A security group that contains additional fields for describing hierarchical relationships between organizations.

**pinned link:** A link that users can attach permanently to the top link bar of a personal site.

**privacy level:** A setting that specifies the category of users who are permitted to view personal information about other users, such as user profile properties, colleagues, or memberships.

**profile site:** A page that can display detailed information about a user by using a URL prefix that can be concatenated with a login name, email address, or GUID that identifies the user.

**quick link:** A URL that is saved by a user for later reference.

**site:** A group of related pages and data within a SharePoint site collection. The structure and content of a site is based on a site definition. Also referred to as SharePoint site and web site.

**site collection administrator:** A user who has administrative permissions for a site collection.

**SOAP:** A lightweight protocol for exchanging structured information in a decentralized, distributed environment. **SOAP** uses XML technologies to define an extensible messaging framework, which provides a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular programming model and other implementation-specific semantics. SOAP 1.2 supersedes SOAP 1.1. See [\[SOAP1.2-1/2003\]](#).

**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:** A container for error and status information within a SOAP message. See [\[SOAP1.2-1/2007\]](#) section 5.4 for more information.

**term set:** A collection of terms that are arranged into and stored as a hierarchy or a flat list.

**Uniform Resource Locator (URL):** A string of characters in a standardized format that identifies a document or resource on the World Wide Web. The format is as specified in [\[RFC1738\]](#).

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

**user profile policy:** A subset of the user profile privacy policy that governs interactions with user profiles.

**user profile privacy policy:** A set of rules that governs all interactions with user profiles.

**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.

**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 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](mailto: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>

[RFC2396] Berners-Lee, T., Fielding, R., and Masinter, L., "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, August 1998, <http://www.rfc-editor.org/rfc/rfc2396.txt>

[RFC2616] Fielding, R., Gettys, J., Mogul, J., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, <http://www.rfc-editor.org/rfc/rfc2616.txt>

[SOAP1.1] Box, D., Ehnebuske, D., Kakivaya, G., et al., "Simple Object Access Protocol (SOAP) 1.1", W3C Note, May 2000, <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>

[SOAP1.2-1/2007] Gudgin, M., Hadley, M., Mendelsohn, N., et al., "SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)", W3C Recommendation, April 2007, <http://www.w3.org/TR/2007/REC-soap12-part1-20070427/>

[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/>

## 1.2.2 Informative References

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000, <http://www.rfc-editor.org/rfc/rfc2818.txt>

[SOAP1.2-2/2007] Gudgin, M., Hadley, M., Mendelsohn, N., et al., "SOAP Version 1.2 Part 2: Adjuncts (Second Edition)", W3C Recommendation, April 2007, <http://www.w3.org/TR/2007/REC-soap12-part2-20070427>

## 1.3 Protocol Overview (Synopsis)

This protocol enables a protocol client to retrieve **user profile** data, and data in common among users, as well as to manage other user profile properties, such as links, **group(2) memberships**, and **colleagues**.

This protocol can be used to manage an enterprise social network and to enable users to search for and find people within their **organization** for communication purposes.

The protocol server exposes the personal information of each user as a user profile containing a set of properties. Profile properties are name/value pairs with a corresponding **privacy level**. A user can restrict access and viewing of a profile property to the following access levels:

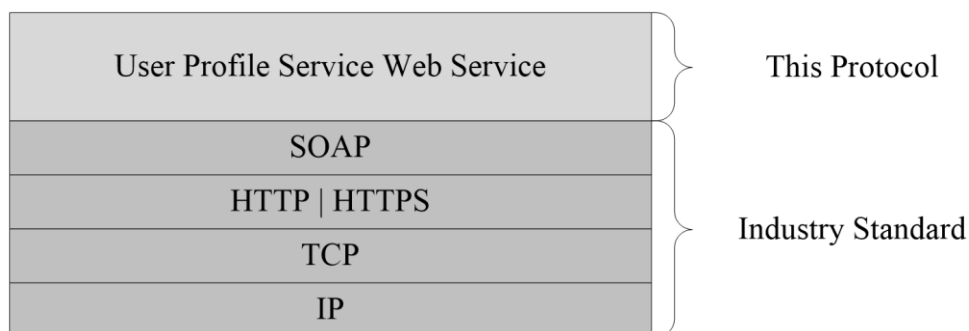
- Only Me (Private)
- My Manager (Manager)
- My Workgroup (Organization)
- My Colleagues (Contacts)
- Everyone (Public)

The protocol server can also enforce a default privacy policy for a property set by a **site collection administrator**.

## 1.4 Relationship to Other Protocols

This protocol uses the **SOAP** message protocol for formatting request and response messages, as described in [\[SOAP1.1\]](#), [\[SOAP1.2-1/2007\]](#) and [\[SOAP1.2-2/2007\]](#). It transmits those messages by using **HTTP**, as described in [\[RFC2616\]](#), or **Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)**, as described in [\[RFC2818\]](#).

The following diagram shows the underlying messaging and transport stack used by the protocol:



## Figure 1: This protocol in relation to other protocols

### 1.5 Prerequisites/Preconditions

This protocol operates against a **site** that is identified by a **URL** that is known by protocol clients. The protocol server endpoint is formed by appending "\_vti\_bin/userprofiles.service.asmx" to the URL of the site, for example [http://www.contoso.com/Repository/\\_vti\\_bin/userprofiles.service.asmx](http://www.contoso.com/Repository/_vti_bin/userprofiles.service.asmx).

This protocol assumes that authentication has been performed by the underlying protocols.

### 1.6 Applicability Statement

This protocol is designed to work with up to 5 million **user profiles**. For each user profile, it supports up to 100 user profile properties.

### 1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- **Supported Transports:** This protocol uses multiple transports with SOAP as specified in section [2.1](#).

### 1.8 Vendor-Extensible Fields

None.

### 1.9 Standards Assignments

None.

## 2 Messages

### 2.1 Transport

Protocol servers MUST support SOAP over HTTP. Protocol servers SHOULD additionally support SOAP over HTTPS for securing communication with clients.

Protocol messages MUST be formatted as specified either in [\[SOAP1.1\]](#) section 4 or in [\[SOAP1.2-1/2007\]](#) section 5. Protocol server faults MUST be returned either by using HTTP Status Codes as specified in [\[RFC2616\]](#) section 10 or by using **SOAP faults** as specified either in [\[SOAP1.1\]](#) section 4.4 or in [\[SOAP1.2-1/2007\]](#) section 5.4.

### 2.2 Common Message Syntax

This section contains common definitions used by this protocol. The syntax of the definitions uses **XML schema** as defined in [\[XMLSCHEMA1\]](#) and [\[XMLSCHEMA2\]](#), and **WSDL** as defined 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
s1	http://microsoft.com/wsdl/types/	
soap	http://schemas.xmlsoap.org/wsdl/soap/	<a href="#">[SOAP1.1]</a>
s	http://www.w3.org/2001/XMLSchema	<a href="#">[XMLSCHEMA1]</a> <a href="#">[XMLSCHEMA2]</a>
tns	http://microsoft.com/webservices/SharePointPortalServer/UserProfileService	
wsdl	http://schemas.xmlsoap.org/wsdl/	<a href="#">[WSDL]</a>

#### 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
ArrayOfContactData	A set of <b>ContactData</b> (section <a href="#">2.2.4.11</a> ) elements.
ArrayOfDouble	A set of double elements.
ArrayOfMembershipData	A set of <b>MembershipData</b> (section <a href="#">2.2.4.13</a> ) elements.
ArrayOfOrganizationProfileData	A set of <b>OrganizationProfileData</b> (section <a href="#">2.2.4.14</a> ) elements.
ArrayOfPinnedLinkData	A set of <b>PinnedLinkData</b> (section <a href="#">2.2.4.15</a> ) elements.
ArrayOfPropertyData	A set of <b>PropertyData</b> (section <a href="#">2.2.4.16</a> ) elements.
ArrayOfPropertyInfo	A set of <b>PropertyInfo</b> (section <a href="#">2.2.4.17</a> ) elements.
ArrayOfQuickLinkData	A set of <b>QuickLinkData</b> (section <a href="#">2.2.4.18</a> ) elements.
ArrayOfString	A set of string elements.
ArrayOfValueData	A set of <b>ValueData</b> (section <a href="#">2.2.4.19</a> ) elements.
ContactData	Identifies a <b>contact (2)</b> of the <b>user profile</b> .
MemberGroupData	Identifies a <b>member group</b> .
MembershipData	Identifies a <b>membership</b> .
OrganizationProfileData	An <b>organization</b> .
PinnedLinkData	A <b>pinned link</b> .
PropertyData	A property of a user profile object.
PropertyInfo	Describes the attributes of a property.
QuickLinkData	A <b>quick link</b> .
ValueData	This type is one value for a given user profile property.

### 2.2.4.1 ArrayOfContactData

The **ArrayOfContactData** complex type contains a set of **ContactData** (section [2.2.4.11](#)) elements.

```
<s:complexType name="ArrayOfContactData">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="unbounded" name="ContactData" nillable="true"
type="tns:ContactData" />
  </s:sequence>
</s:complexType>
```

**ContactData:** A single **contact (2)**.

### 2.2.4.2 ArrayOfDouble

The **ArrayOfDouble** complex type contains a set of double elements.

```
<s:complexType name="ArrayOfDouble">
```

```
<s:sequence>
  <s:element minOccurs="0" maxOccurs="unbounded" name="double" type="s:double" />
</s:sequence>
</s:complexType>
```

**double:** A single element in the array of doubles.

### 2.2.4.3 ArrayOfMembershipData

The **ArrayOfMembershipData** complex type contains a set of **MembershipData** (section [2.2.4.13](#)) elements.

```
<s:complexType name="ArrayOfMembershipData">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="unbounded" name="MembershipData" nillable="true"
type="tns:MembershipData" />
  </s:sequence>
</s:complexType>
```

**MembershipData:** A single **membership**.

### 2.2.4.4 ArrayOfOrganizationProfileData

The **ArrayOfOrganizationProfileData** complex type contains a set of **OrganizationProfileData** (section [2.2.4.14](#)) elements.

```
<s:complexType name="ArrayOfOrganizationProfileData">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="unbounded" name="OrganizationProfileData"
nillable="true" type="tns:OrganizationProfileData" />
  </s:sequence>
</s:complexType>
```

**OrganizationProfileData:** A single **organization** profile.

### 2.2.4.5 ArrayOfPinnedLinkData

The **ArrayOfPinnedLinkData** complex type contains a set of **PinnedLinkData** (section [2.2.4.15](#)) elements.

```
<s:complexType name="ArrayOfPinnedLinkData">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="unbounded" name="PinnedLinkData" nillable="true"
type="tns:PinnedLinkData" />
  </s:sequence>
</s:complexType>
```

**PinnedLinkData:** A single **pinned link**.

### 2.2.4.6 ArrayOfPropertyData

The **ArrayOfPropertyData** complex type contains a set of **PropertyData** (section [2.2.4.16](#)) elements.

```

<s:complexType name="ArrayOfPropertyData">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="unbounded" name="PropertyData" nillable="true"
type="tns:PropertyData" />
  </s:sequence>
</s:complexType>

```

**PropertyData:** A single property.

#### 2.2.4.7 ArrayOfPropertyInfo

The **ArrayOfPropertyInfo** complex type contains a set of **PropertyInfo** (section [2.2.4.17](#)) elements.

```

<s:complexType name="ArrayOfPropertyInfo">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="unbounded" name="PropertyInfo" nillable="true"
type="tns:PropertyInfo" />
  </s:sequence>
</s:complexType>

```

**PropertyInfo:** Used to represent schema information about a property.

#### 2.2.4.8 ArrayOfQuickLinkData

The **ArrayOfQuickLinkData** complex type contains a set of **QuickLinkData** (section [2.2.4.18](#)) elements.

```

<s:complexType name="ArrayOfQuickLinkData">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="unbounded" name="QuickLinkData" nillable="true"
type="tns:QuickLinkData" />
  </s:sequence>
</s:complexType>

```

**QuickLinkData:** A single **quick link**.

#### 2.2.4.9 ArrayOfString

The **ArrayOfString** complex type is used to represent a set of strings.

```

<s:complexType name="ArrayOfString">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="unbounded" name="string" nillable="true"
type="s:string" />
  </s:sequence>
</s:complexType>

```

**string:** A single element in the array of strings.

#### 2.2.4.10 ArrayOfValueData

The **ArrayOfValueData** complex type contains a set of **ValueData** (section [2.2.4.19](#)) elements.

```

<s:complexType name="ArrayOfValueData">
  <s:sequence>

```

```

    <s:element minOccurs="0" maxOccurs="unbounded" name="ValueData" nillable="true"
type="tns:ValueData" />
  </s:sequence>
</s:complexType>

```

**ValueData:** Represents a single value.

### 2.2.4.11 ContactData

The **ContactData** complex type contains contact information about a **user profile**.

```

<s:complexType name="ContactData">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="AccountName" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="Privacy" type="tns:Privacy" />
    <s:element minOccurs="0" maxOccurs="1" name="Name" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="IsInWorkGroup" type="s:boolean" />
    <s:element minOccurs="0" maxOccurs="1" name="Group" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="Email" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="Title" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="Url" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="UserProfileID" type="s1:guid" />
    <s:element minOccurs="1" maxOccurs="1" name="ID" type="s:long" />
  </s:sequence>
</s:complexType>

```

**AccountName:** A string that specifies the **login name** of the user profile. This value MUST be less than or equal to 400 characters in length.

**Privacy:** The **privacy level** for the **contact (2)** association with the user profile. This MUST be a value in the **Privacy** (section [2.2.5.4](#)) enumeration.

**Name:** A string that specifies the name of the user.

**IsInWorkGroup:** A Boolean value that specifies whether the user specified by this element belongs to the same workgroup as the associated user profile. If associated user profile is in the same workgroup as the user specified in this element the server MUST return **true**.

**Group:** A string that specifies the **group (2)** this contact (1) belongs to.

**Email:** A string that specifies the e-mail address for the user.

**Title:** A string that specifies the title for the user.

**Url:** The **URL** for the user.

**UserProfileID:** The **GUID** of the user.

**ID:** A long that specifies the unique identifier for the user.

### 2.2.4.12 MemberGroupData

The **MemberGroupData** complex type identifies a **member group**.

```

<s:complexType name="MemberGroupData">
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="SourceInternal" type="s1:guid" />
    <s:element minOccurs="0" maxOccurs="1" name="SourceReference" type="s:string" />
  </s:sequence>

```

</s:complexType>

**SourceInternal:** A **GUID** that specifies the original source of members for the member group. If the original source was from a **distribution list** or **site** then the GUID MUST be one of the values listed in the following table. Otherwise the GUID MUST correspond to the user-defined GUID for that source.

GUID value	Description
A88B9DCB-5B82-41E4-8A19-17672F307B95	The member group was originally populated from a distribution list.
8BB1220F-DE8B-4771-AC3A-0551242CF2BD	The member group was originally populated from a site.

**SourceReference:** A string that uniquely identifies the member group.

### 2.2.4.13 MembershipData

The **MembershipData** complex type represents a membership.

```
<s:complexType name="MembershipData">
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="Source" type="tns:MembershipSource" />
    <s:element minOccurs="0" maxOccurs="1" name="MemberGroup" type="tns:MemberGroupData" />
    <s:element minOccurs="0" maxOccurs="1" name="Group" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="DisplayName" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="Privacy" type="tns:Privacy" />
    <s:element minOccurs="0" maxOccurs="1" name="MailNickname" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="Url" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="ID" type="s:long" />
    <s:element minOccurs="1" maxOccurs="1" name="MemberGroupID" type="s:long" />
  </s:sequence>
</s:complexType>
```

**Source:** This value specifies the original source of members for the **member group**. It MUST be a value specified in **MembershipSource** (section [2.2.5.3](#)).

**MemberGroup:** A **MemberGroupData** (section [2.2.4.12](#)) type that identifies the member group involved in the **membership**.

**Group:** A string that specifies the **group (2)** of this membership. This value MUST be less than or equal to 50 characters in length.

**DisplayName:** A string that specifies the display name of the member group involved in the membership.

**Privacy:** Specifies the category of users who can view this membership. This MUST be a value in the **Privacy** (section [2.2.5.4](#)) enumeration.

**MailNickname:** A string that specifies the e-mail alias of the member group involved in the membership.

**Url:** The **URL** of the member group involved in the membership.

**ID:** A long that uniquely identifies this membership.

**MemberGroupID:** A long that uniquely identifies the member group involved in the membership.

#### 2.2.4.14 OrganizationProfileData

The **OrganizationProfileData** complex type represents an **organization**.

```
<s:complexType name="OrganizationProfileData">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="DisplayName" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="RecordID" type="s:long" />
  </s:sequence>
</s:complexType>
```

**DisplayName:** A string that specifies the display text for the organization.

**RecordID:** A long that uniquely identifies the organization.

#### 2.2.4.15 PinnedLinkData

The **PinnedLinkData** complex type represents a **pinned link**.

```
<s:complexType name="PinnedLinkData">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="Name" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="Url" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="ID" type="s:long" />
  </s:sequence>
</s:complexType>
```

**Name:** A string that specifies the display text for the pinned link.

**Url:** The **URL** of the pinned link.

**ID:** A long that uniquely identifies the pinned link.

#### 2.2.4.16 PropertyData

The **PropertyData** complex type contains information about a **user profile** property.

```
<s:complexType name="PropertyData">
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="IsPrivacyChanged" type="s:boolean" />
    <s:element minOccurs="1" maxOccurs="1" name="IsValueChanged" type="s:boolean" />
    <s:element minOccurs="0" maxOccurs="1" name="Name" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="Privacy" type="tns:Privacy" />
    <s:element minOccurs="0" maxOccurs="1" name="Values" type="tns:ArrayOfValueData" />
  </s:sequence>
</s:complexType>
```

**Name:** A string that specifies the name of the user profile property.

**Privacy:** The **privacy level** of the user profile property. This MUST be a value in the **Privacy** (section [2.2.5.4](#)) enumeration.

**Values:** A set of **ValueData** (section [2.2.4.19](#)) elements.

**IsPrivacyChanged:** A Boolean value indicating whether or not the privacy value for this user profile property has changed. When a **PropertyData** element is returned from the protocol server, **IsPrivacyChanged** MUST be **false**.

**IsValueChanged:** A Boolean value indicating whether or not the values for this user profile property have changed. When a **PropertyData** element is returned from the protocol server, **IsValueChanged** MUST be **false**.

### 2.2.4.17 PropertyInfo

The **PropertyInfo** complex type is used to represent schema information about a property.

```
<s:complexType name="PropertyInfo">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="Name" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="Description" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="DisplayOrder" type="s:int" />
    <s:element minOccurs="0" maxOccurs="1" name="MaximumShown" type="s:int" />
    <s:element minOccurs="0" maxOccurs="1" name="IsAdminEditable" type="s:boolean" />
    <s:element minOccurs="0" maxOccurs="1" name="IsSearchable" type="s:boolean" />
    <s:element minOccurs="0" maxOccurs="1" name="IsSystem" type="s:boolean" />
    <s:element minOccurs="0" maxOccurs="1" name="ManagedPropertyName" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="DisplayName" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="Type" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="AllowPolicyOverride" type="s:boolean" />
    <s:element minOccurs="1" maxOccurs="1" name="DefaultPrivacy" type="tns:Privacy" />
    <s:element minOccurs="1" maxOccurs="1" name="IsAlias" type="s:boolean" />
    <s:element minOccurs="1" maxOccurs="1" name="IsColleagueEventLog" type="s:boolean" />
    <s:element minOccurs="1" maxOccurs="1" name="IsRequired" type="s:boolean" />
    <s:element minOccurs="1" maxOccurs="1" name="IsUserEditable" type="s:boolean" />
    <s:element minOccurs="1" maxOccurs="1" name="IsVisibleOnEditor" type="s:boolean" />
    <s:element minOccurs="1" maxOccurs="1" name="IsVisibleOnViewer" type="s:boolean" />
    <s:element minOccurs="1" maxOccurs="1" name="IsReplicable" type="s:boolean" />
    <s:element minOccurs="1" maxOccurs="1" name="UserOverridePrivacy" type="s:boolean" />
    <s:element minOccurs="1" maxOccurs="1" name="Length" type="s:int" />
    <s:element minOccurs="1" maxOccurs="1" name="IsImported" type="s:boolean" />
    <s:element minOccurs="1" maxOccurs="1" name="IsMultiValue" type="s:boolean" />
    <s:element minOccurs="1" maxOccurs="1" name="ChoiceType" type="tns:ChoiceTypes" />
    <s:element minOccurs="0" maxOccurs="1" name="TermSetId" nillable="true" type="s1:guid" />
  </s:sequence>
</s:complexType>
```

**Name:** A string that specifies the name of the property.

**Description:** A string that specifies a description of the property.

**DisplayOrder:** An integer that specifies the index at which this property is normally displayed.

**MaximumShown:** An integer that specifies the maximum number of values shown for this property.

**IsAdminEditable:** A Boolean that if set to **true** specifies that this property can be edited by an administrator and is set to **false** otherwise.

**IsSearchable:** A Boolean that if set to **true** specifies that this property is searchable and is set to **false** otherwise.

**IsSystem:** A Boolean that if set to **true** specifies that this is a core system property and is set to **false** otherwise.

**ManagedPropertyName:** If specified, a string that is used in **URL** encodings when searching for other properties with the same value.

**DisplayName:** A string that specifies the display name of the property. **Type:** A string that specifies the value specifying the type of the **user profile** property.

**AllowPolicyOverride:** A Boolean that if set to **true** specifies that the **site collection administrator** is allowed to override the default policy setting assigned to the **user profile privacy policy** and is set to **false** otherwise.

**DefaultPrivacy:** A value indicating the default privacy setting of the property which MUST be a value listed in the **Privacy** (section [2.2.5.4](#)) enumeration.

**IsAlias:** A Boolean that if set to **true** specifies that this property serves as an alias property of the user profile for user search purposes and is set to **false** otherwise.

**IsColleagueEventLog:** A Boolean that if set to **true** specifies that this property is displayed on the **Colleague Tracker Web Part** and is set to **false** otherwise

**IsRequired:** A Boolean that if set to **true** specifies that this property is required when saving its value to a user profile and is set to **false** otherwise.

**IsUserEditable:** A Boolean value that if set to **true** specifies that this property is editable and is set to **false** otherwise

**IsVisibleOnEditor:** A Boolean that if set to **true** specifies that this property is visible when a user edits his or her user profile from his or her **profile site** and is set to **false** otherwise.

**IsVisibleOnViewer:** A Boolean value that if set to **true** specifies this property is visible on the default profile site and is set to **false** otherwise.

**IsReplicable:** A Boolean value that if set to **true** specifies this property is replicable and is set to **false** otherwise.

**UserOverridePrivacy:** A Boolean that if set to **true** specifies that the system enables the user to override the default protection level assigned to the privacy policy and is set to **false** otherwise.

**Length:** An integer that specifies the maximum length of the value of the property.

**IsImported:** A Boolean value that if set to **true** specifies that this property is imported to the protocol server from an external data source and is set to **false** otherwise.

**IsMultiValue:** A Boolean that if set to **true** specifies that the property can have multiple values and is set to **false** otherwise.

**ChoiceType:** A **ChoiceTypes** (section [2.2.5.1](#)) value which indicates the state of the property's list of choices.

**TermSetID:** If specified, a **GUID** that corresponds to a taxonomy **term set** that backs the value of the property.

## 2.2.4.18 QuickLinkData

The **QuickLinkData** complex type represents a **quick link**.

```
<s:complexType name="QuickLinkData">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="Name" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="Group" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="Privacy" type="tns:Privacy" />
    <s:element minOccurs="0" maxOccurs="1" name="Url" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="ID" type="s:long" />
  </s:sequence>
</s:complexType>
```

**Name:** A string that specifies the display text for the quick link.



**Group:** A string that specifies the **group (2)** this quick link belongs to.

**Privacy:** Specifies the category of users who can view this quick link. This MUST be a value in the **Privacy** (section [2.2.5.4](#)) enumeration.

**Url:** The **URL** of the quick link.

**ID:** A long that uniquely identifies the quick link.

### 2.2.4.19 ValueData

The **ValueData** complex type is used to represent a value for a given **user profile** property.

```
<s:complexType name="ValueData">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="Value" type="s:string" />
  </s:sequence>
</s:complexType>
```

**Value:** A string element contains the value of a user profile property.

## 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
ChoiceTypes	Used to indicate the state of a list of choices for a property.
guid	A simple type used to represent a <b>GUID</b> .
MembershipSource	Specifies the source of a <b>member group</b> involved in a particular <b>membership</b> .
Privacy	Specifies the set of users who are allowed to access a protected resource.
SuggestionType	Specifies the type of suggestion (for example, <b>keyword</b> suggestion vs. <b>colleague</b> suggestion).

### 2.2.5.1 ChoiceTypes

The **ChoiceTypes** simple type is used to indicate the state of a list of choices for a property.

```
<s:simpleType name="ChoiceTypes">
  <s:restriction base="s:string">
    <s:enumeration value="Off"/>
    <s:enumeration value="None"/>
    <s:enumeration value="Open"/>
    <s:enumeration value="Closed"/>
  </s:restriction>
</s:simpleType>
```

Value	Meaning
Off	The property does not use a list of choices.
None	The property is not using a list of choices, but a list of choices can be added by a <b>site collection administrator</b> .
Open	The property is using a list of choices and users can add items to the list of choices.
Closed	The property uses a list of choices but users cannot edit the choices. A site collection administrator has full control over the choices for the property.

### 2.2.5.2 guid

The **guid** simple type is used to represent a **GUID**.

```
<s:simpleType name="guid">
  <s:restriction base="s:string">
    <s:pattern value="[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}"/>
  </s:restriction>
</s:simpleType>
```

### 2.2.5.3 MembershipSource

The **MembershipSource** simple type is an enumeration used to specify the source of a **member group** involved in a particular **membership**.

```
<s:simpleType name="MembershipSource">
  <s:restriction base="s:string">
    <s:enumeration value="DistributionList"/>
    <s:enumeration value="SharePointSite"/>
    <s:enumeration value="Other"/>
  </s:restriction>
</s:simpleType>
```

Value	Meaning
DistributionList	The member group was originally populated from a <b>distribution list</b> .
SharePointSite	The member group was originally populated from a <b>site</b> .
Other	The member group was originally populated from a source external to a site server.

## 2.2.5.4 Privacy

The **Privacy** simple type specifies the set of users who are allowed to access a protected resource.

```
<s:simpleType name="Privacy">
  <s:restriction base="s:string">
    <s:enumeration value="Public"/>
    <s:enumeration value="Contacts"/>
    <s:enumeration value="Organization"/>
    <s:enumeration value="Manager"/>
    <s:enumeration value="Private"/>
    <s:enumeration value="NotSet"/>
  </s:restriction>
</s:simpleType>
```

Value	Meaning
Public	All users are allowed to access the resource.
Contacts	The only users allowed to access the resource are the owner of the resource and the owner's <b>colleagues</b> .
Organization	The only users allowed to access the resource are the owner of the resource and the owner's workgroup colleagues.
Manager	The only two users allowed to access the resource are the owner of the resource and the owner's manager.
Private	The only user allowed to access the resource is the owner of the resource.
NotSet	The security settings for the item have not been overridden by the user that created the resource. The settings as defined in the <b>user profile policy</b> are applied.

## 2.2.5.5 SuggestionType

The **SuggestionType** simple type is an enumeration that specifies the type of suggestion.

```
<s:simpleType name="SuggestionType">
  <s:restriction base="s:string">
    <s:enumeration value="Colleague" />
    <s:enumeration value="Keyword" />
  </s:restriction>
</s:simpleType>
```

Value	Meaning
Colleague	The <b>Suggestion</b> corresponds to a <b>Colleague</b> that is specified by a string with either an <b>accountName</b> or <b>email</b> value.
Keyword	The <b>Suggestion</b> corresponds to a <b>keyword</b> that is specified by a string.

### 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.

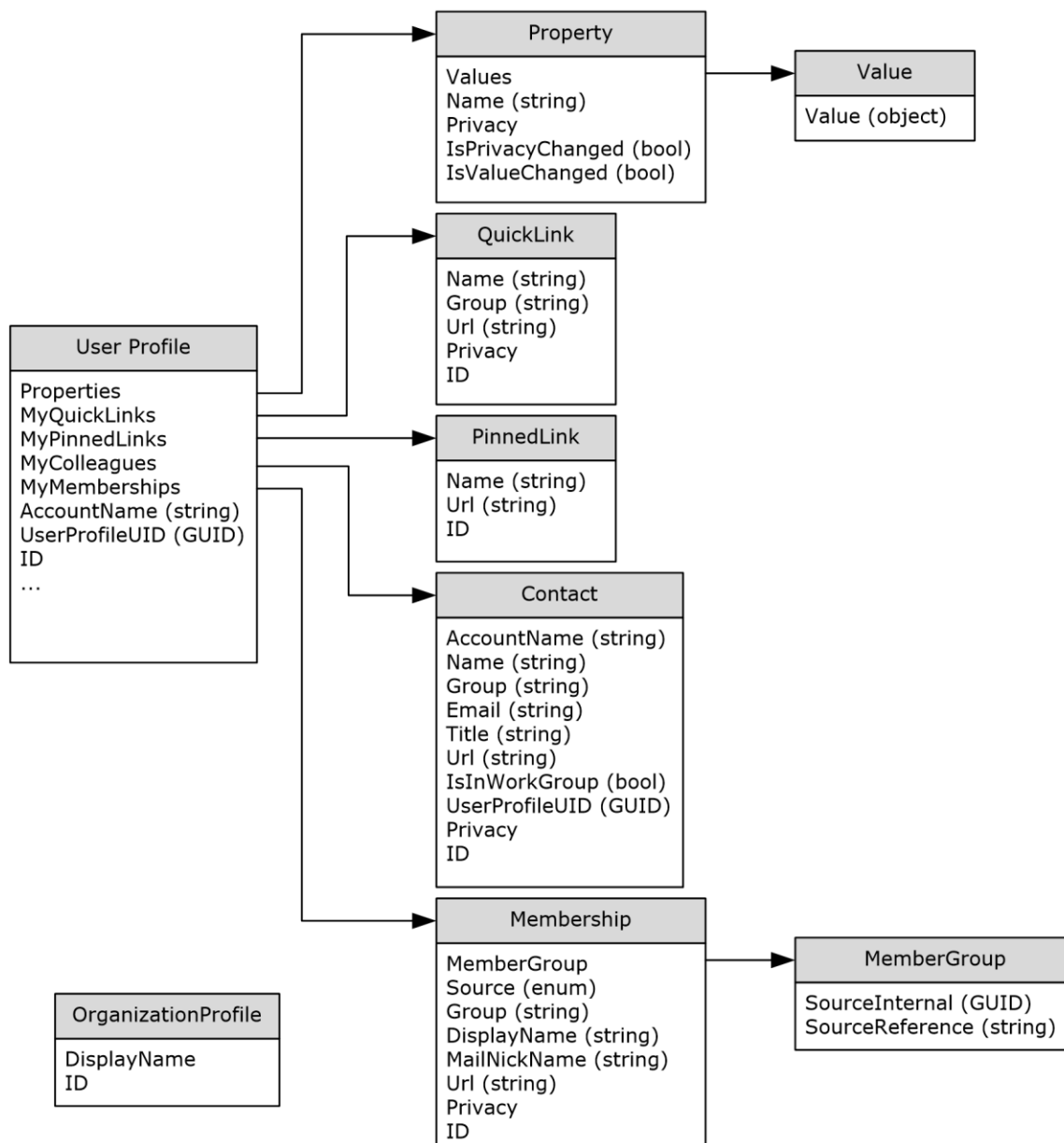
## **3 Protocol Details**

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 made by the higher-layer protocol or application are passed directly to the transport, and the results returned by the transport are passed directly back to the higher-layer protocol or application.

### **3.1 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.



**Figure 2: Protocol server abstract data model**

In the previous figure, each table specifies a type of entity in the model. Each arrow specifies that one type of entity contains one or more references to another.

**UserProfile Entity:** An entity that corresponds to a **user profile** available on the protocol server. A **UserProfile** entity contains references to properties, **quick links**, **pinned links**, **colleagues**, and **memberships**. This entity can also contain various other pieces of data associated with the user.

**Properties Field:** References a collection of **Property** data model entities.

**MyQuickLinks Field:** References a collection of **QuickLink** data model entities.

**MyPinnedLinks Field:** References a collection of **PinnedLink** data model entities.

**MyColleagues Field:** References a collection of **contact (2)** data model entities.

**MyMemberships Field:** References a collection of **Membership** data model entities.

**AccountName Field:** A string value that serves as the **login name** for the account associated with this **UserProfile** entity.

**UserProfileUID Field:** A **GUID** value that uniquely identifies the user profile.

**ID Field:** A unique identifier that identifies the **UserProfile** entity.

**Property Entity:** An entity that corresponds to a property. A **Property** entity belongs to a **UserProfile** entity and represents a property of a user profile. A **Property** entity implements the concept of associating one or more values with a given property name.

**Values Field:** References a collection of **Value** data model entities. These entities serve as the values for this **Property** entity.

**Name Field:** A string value that serves as the name for this **Property** entity.

**Privacy Field:** An enumerated value corresponding to the **privacy level** for this **Property** entity. Conceptually, this **Privacy** value determines who has the rights to view this property.

**IsPrivacyChanged Field:** A Boolean value indicating whether or not the **Privacy** for this **Property** entity has changed. This can be used for caching to prevent excess data queries.

**IsValueChanged Field:** A Boolean value indicating whether or not the **Values** for this **Property** entity have changed. This can be used for caching to prevent excess data queries.

**Value Entity:** An entity that contains a value of a **Property** data model entity.

**Value Field:** An object that serves as the value for this **Value** entity.

**QuickLink Entity:** An entity that corresponds to a quick link, which is associated with a **UserProfile** data model entity.

**Name Field:** A string value that serves as the name for this **QuickLink** entity.

**Group Field:** A string value corresponding to the **group (2)** for this **QuickLink** entity.

**Url Field:** A string value corresponding to the **URL** for this **QuickLink** entity.

**Privacy Field:** An enumerated value corresponding to the privacy level for this **QuickLink** entity. Conceptually, this **Privacy** value determines who has the rights to view this quick link.

**ID Field:** A unique identifier that identifies the **QuickLink** entity.

**PinnedLink Entity:** An entity that corresponds to a pinned link, which is associated with a **UserProfile** data model entity.

**Name Field:** A string value that serves as the name for this **PinnedLink** entity.

**Url Field:** A string value corresponding to the URL for this **PinnedLink** entity.

**ID Field:** A unique identifier that identifies the **PinnedLink** entity.

**Contact Entity:** An entity that contains the necessary data to store contact (2) information for another user who is a colleague of the associated **UserProfile** data model entity. Conceptually, each contact (1) entity serves as a one-way link from the specified **UserProfile** entity to contact (1) .

**AccountName Field:** A string value corresponding to the **AccountName** of the colleague's **UserProfile** entity.

**Name Field:** A string value that serves as the name for this contact (1) entity.

**Group Field:** A string value corresponding to the group (1) for this contact (1) entity.

**Email Field:** A string value corresponding to the e-mail address for this contact (1) entity.

**Title Field:** A string value corresponding to the title for this contact (1) entity.

**Url Field:** A string value corresponding to the URL for this contact (1) entity.

**IsInWorkGroup Field:** A Boolean value indicating whether or not the colleague for this contact (1) entity is in the associated user profile's workgroup. If associated user profile is in the same workgroup as the colleague for this contact (1) entity the server MUST return **true**.

**UserProfileUID Field:** A GUID value that uniquely identifies the user profile of the colleague.

**Privacy Field:** An enumerated value corresponding to the privacy level for this contact (1) entity. Conceptually, this **Privacy** value determines who has the rights to view this colleague relationship.

**ID Field:** A unique identifier that identifies the contact (1) entity.

**Membership Entity:** An entity that contains the necessary data to store membership information. Conceptually, a **Membership** entity serves as a one-way link from a **UserProfile** entity to a **MemberGroup** entity.

**MemberGroup Field:** A reference to this **Membership** entity's **MemberGroup** data model entity.

**Source Field:** An enumerated value corresponding to the source of members for this **Membership** entity. Conceptually, this **Source** value indicates whether this membership is to a **distribution list**, **site**, or "other".

**Group Field:** A string value corresponding to the group (2) for this **Membership** entity.

**DisplayName Field:** A string value corresponding to the display name for this **Membership** entity.

**MailNickName Field:** A string value corresponding to the e-mail alias for this **Membership** entity.

**Url Field:** A string value corresponding to the URL for this **Membership** entity.

**Privacy Field:** An enumerated value corresponding to the privacy level for this contact (1) entity. Conceptually, this **Privacy** value determines who has the rights to view this membership relationship.

**ID Field:** A unique identifier that identifies the **Membership** entity.

**MemberGroup Entity:** An entity that represents a **member group**. Conceptually, this entity is a node for **Membership** entities to reference.

**SourceInternal Field:** A GUID value indicating the original source of members for the member group as specified in **MemberGroupData** (section [2.2.4.12](#)).

**SourceReference Field:** A string that uniquely identifies the member group.

### 3.1.2 Timers

None.

### 3.1.3 Initialization

The following initialization steps MUST be performed:

1. All abstract data model entities contain data MUST be retrieved from persistent storage at initialization time.



2. Each operation within the protocol server MUST begin listening for requests at the respective URL addresses given in the message transport (as specified in section [2.1](#)).

### 3.1.4 Message Processing Events and Sequencing Rules

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

Operation	Description
AddColleague	Updates the specified <b>user profile</b> by adding a <b>colleague</b> association.
AddColleagueWithoutEmailNotification	Updates the specified user profile by adding a colleague association, but the server will not send any email notifications upon successful addition of a new colleague.
AddLink	Updates the specified user profile by adding a <b>quick link</b> .
AddMembership	Updates the specified user profile by adding a <b>membership</b> .
AddPinnedLink	Updates the specified user profile by adding a <b>pinned link</b> .
AddSuggestions <a href="#">&lt;1&gt;</a>	Add a suggestion for the user profile of the <b>current user</b> .
CreateMemberGroup	Creates a new <b>member group</b> .
CreateUserProfileByAccountName	Creates a user profile for the specified user <b>login name</b> .
GetCommonColleagues	Retrieves the common colleagues between two user profiles.
GetCommonManager	Retrieves the lowest level common manager between two user profiles.
GetCommonMemberships	Retrieves the common memberships between two user profiles.
GetInCommon	Retrieves the common memberships, colleagues, and lowest level common manager between two user profiles.
GetProfileSchema <a href="#">&lt;2&gt;</a>	Retrieves schema information about each of the profile properties defined.
GetProfileSchemaNameByAccountName <a href="#">&lt;3&gt;</a>	Retrieves the name of the schema for the user profile of the specified user login name.
GetProfileSchemaNames <a href="#">&lt;4&gt;</a>	Retrieves a list of profile schema names.
GetPropertyChoiceList	Retrieves a list of possible values for a property.
GetUserColleagues	Retrieves the colleagues associated with a user profile.
GetUserLinks	Retrieves the quick links associated with a user profile.
GetUserMemberships	Retrieves the memberships associated with a user profile.

Operation	Description
GetUserOrganizations<5>	Retrieves the <b>organizations</b> associated with a user profile.
GetUserPinnedLinks	Retrieves the pinned links associated with a user profile.
GetUserProfileByGuid	Retrieve a user profile given the <b>GUID</b> of a user.
GetUserProfileByIndex	Retrieves a user profile given a unique identifier.
GetUserProfileByName	Retrieves a user profile given a login name.
GetUserProfileCount	Returns the number of user profiles stored by the protocol server.
GetUserProfileSchema	Retrieves schema information about each of the properties defined.
GetUserPropertyByAccountName<6>	Retrieves a user profile property given a user login name and property name.
ModifyUserPropertyByAccountName	Updates a set of properties for a user.
RemoveAllColleagues	Deletes all colleagues associated with a user profile.
RemoveAllLinks	Deletes all quick links associated with a user profile.
RemoveAllMemberships	Deletes all memberships associated with a user profile.
RemoveAllPinnedLinks	Deletes all pinned links associated with a user profile.
RemoveColleague	Deletes a colleague associated with a user profile.
RemoveLink	Deletes a quick link associated with a user profile.
RemoveMembership	Deletes a membership associated with a user profile.
RemovePinnedLink	Deletes a pinned link associated with a user profile.
UpdateColleaguePrivacy	Updates the <b>privacy level</b> of a colleague association.
UpdateLink	Updates a quick link for the specified user.
UpdateMembershipPrivacy	Updates the privacy level of a membership.
UpdatePinnedLink	Updates a pinned link for the specified user.

### 3.1.4.1 AddColleague

The **AddColleague WSDL** operation is used to update the specified **user profile** by adding a **colleague** association.

The operation of adding a colleague is unidirectional, meaning that when a colleague is added to the specified user, the user profile of the colleague will not be updated.

```

<wsdl:operation name="AddColleague">
  <wsdl:input message="tns:AddColleagueSoapIn"/>
  <wsdl:output message="tns:AddColleagueSoapOut"/>
</wsdl:operation>

```

The protocol client sends an **AddColleagueSoapIn** request WSDL message and the protocol server responds with an **AddColleagueSoapOut** response WSDL message. If the user is not authorized to administer the user profile service and is not updating his or her own user profile, the protocol server MUST send a **SOAP fault** message. Additionally, if the user does not have the Use Personalization Features permission, the protocol server MUST send a SOAP fault message. The Use Personalization Features permission is defined for users in the User Profile Service Application and governs the user's access to view other's profiles and view/edit their own profile.



**Figure 3: AddColleague operation**

The **AddColleagueWithoutEmailNotification** WSDL operation behaves identically to the **AddColleague** WSDL operation, except that the server will not send any email notifications upon successful addition of a new colleague.

### 3.1.4.1.1 Messages

The following **WSDL** message definitions are specific to this operation.

#### 3.1.4.1.1.1 AddColleagueSoapIn

The **AddColleagueSoapIn** message is the request **WSDL** message for the **AddColleague** WSDL operation.

The **SOAP action** value of the message is defined as:

```
http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/AddColleague
```

The **SOAP body** contains an **AddColleague** element.

#### 3.1.4.1.1.2 AddColleagueSoapOut

The **AddColleagueSoapOut** message is the response **WSDL** message for the **AddColleague** WSDL operation.

The **SOAP body** contains an **AddColleagueResponse** element.

### 3.1.4.1.2 Elements

The following **XML schema** element definitions are specific to this operation.

#### 3.1.4.1.2.1 AddColleague

The **AddColleague** element is the body of the request **WSDL** message for the **AddColleague** WSDL operation.

```

<s:element name="AddColleague">
  <s:complexType>
    <s:sequence>
      <s:element name="accountName" type="s:string" minOccurs="0" maxOccurs="1"/>
      <s:element name="colleagueAccountName" type="s:string" minOccurs="0"
        maxOccurs="1"/>
      <s:element name="group" type="s:string" minOccurs="0" maxOccurs="1"/>
      <s:element minOccurs="1" maxOccurs="1" name="privacy" type="tns:Privacy"/>
      <s:element minOccurs="1" maxOccurs="1" name="isInWorkGroup" type="s:boolean"/>
    </s:sequence>
  </s:complexType>
</s:element>

```

**accountName:** A string that specifies the **login name** of the **user profile** to be updated. This value MUST be less than or equal to 400 characters in length. If this element is not specified or does not have a value, the protocol server MUST use the user profile of the **current user**. The protocol server MUST treat **accountName** in a case insensitive manner for retrieval of the user profile.

If **accountName** does not correspond to an existing user profile, the protocol server MUST send a **SOAP fault** message.

**colleagueAccountName:** A string that specifies the login name of the user profile of the **colleague** to be added to the user profile specified by **accountName**. This value MUST be less than or equal to 400 characters in length. If this element is not specified or does not have a value, the protocol server MUST use the user profile of the current user, and return a SOAP fault message if the current user does not have a user profile. The protocol server MUST treat **colleagueAccountName** in a case insensitive manner for retrieval of the user profile.

If **colleagueAccountName** does not correspond to an existing user profile, the protocol server MUST send a SOAP fault message.

The user specified by **colleagueAccountName** MUST NOT be the same as the user specified by **accountName**.

If the user specified by **accountName** and the user specified by **colleagueAccountName** are already colleagues, the protocol server MUST send a SOAP fault message.

If the **accountName** is not specified or does not have a value and the **colleagueAccountName** is not specified or does not have a value, the protocol server MUST send a SOAP fault message.

**group:** A string that refers to a **group (2)**. This value MUST be less than or equal to 50 characters in length.

**privacy:** A value that MUST correspond to a **Privacy** (section [2.2.5.4](#)) enumeration. It specifies the set of users who can view this colleague association. This value MUST NOT be "NotSet" as specified in **Privacy** (section 2.2.5.4). If this value is set to "NotSet" the server MUST send a SOAP fault message.

**isInWorkGroup:** A Boolean value which, when **true**, the protocol server MUST allow the user specified in **colleagueAccountName** to see user profile information that falls within the **organization privacy level** for the user specified by **accountName**.

### 3.1.4.1.2.2 AddColleagueResponse

The **AddColleagueResponse** element is the body of the response **WSDL** message for the **AddColleague** WSDL operation.

```

<s:element name="AddColleagueResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="AddColleagueResult" type="tns:ContactData" minOccurs="0"
        maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>

```

**AddColleagueResult:** A **ContactData** (section [2.2.4.11](#)) element that corresponds to the created **colleague** association.

### 3.1.4.2 AddColleagueWithoutEmailNotification

The **AddColleagueWithoutEmailNotification** WSDL operation is used to update the specified **user profile** by adding a **colleague** association.

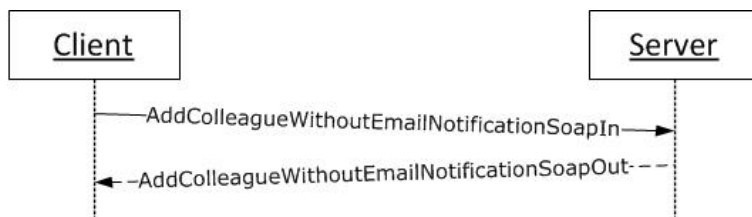
The operation of adding a colleague is unidirectional, meaning that when a colleague is added to the specified user, the user profile of the colleague will not be updated.

```
<wsdl:operation name="AddColleagueWithoutEmailNotification">
  <wsdl:input message="tns:AddColleagueWithoutEmailNotificationSoapIn" />
  <wsdl:output message="tns:AddColleagueWithoutEmailNotificationSoapOut" />
</wsdl:operation>
```

The protocol client sends an **AddColleagueWithoutEmailNotificationSoapIn** request WSDL message and the protocol server responds with an **AddColleagueWithoutEmailNotificationSoapOut** response WSDL message.

If the user is not authorized to administer the user profile service and is not updating his or her own user profile, the protocol server MUST send a **SOAP fault** message.

Additionally, if the user does not have the Use Personalization Features permission, the protocol server MUST send a SOAP fault message. The Use Personalization Features permission is defined for users in the User Profile Service Application and governs the user's access to view other's profiles and view/edit their own profile.



**Figure 4: AddColleagueWithoutEmailNotification operation**

The **AddColleagueWithoutEmailNotification** WSDL operation behaves identically to the **AddColleague** WSDL operation, except that the server will not send any email notifications upon successful addition of a new colleague.

#### 3.1.4.2.1 Messages

The following **WSDL** message definitions are specific to this operation.

##### 3.1.4.2.1.1 AddColleagueWithoutEmailNotificationSoapIn

The **AddColleagueWithoutEmailNotificationSoapIn** message is the request **WSDL** message for the **AddColleagueWithoutEmailNotification** WSDL operation.

The **SOAP action** value of the message is defined as:

```
http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/
AddColleagueWithoutEmailNotification
```

The **SOAP body** contains an **AddColleagueWithoutEmailNotification** element.

##### 3.1.4.2.1.2 AddColleagueWithoutEmailNotificationSoapOut

The **AddColleagueWithoutEmailNotificationSoapOut** message is the response **WSDL** message for the **AddColleagueWithoutEmailNotification** WSDL operation.

The **SOAP body** contains an **AddColleagueWithoutEmailNotificationResponse** element.

### 3.1.4.2.2 Elements

The following **XML schema** element definitions are specific to this operation.

#### 3.1.4.2.2.1 AddColleagueWithoutEmailNotification

The **AddColleagueWithoutEmailNotification** element is the body of the request **WSDL** message for the **AddColleagueWithoutEmailNotification** WSDL operation.

```
<s:element name="AddColleagueWithoutEmailNotification">
  <s:complexType>
    <s:sequence>
      <s:element name="accountName" type="s:string" minOccurs="0" maxOccurs="1"/>
      <s:element name="colleagueAccountName" type="s:string" minOccurs="0"
        maxOccurs="1"/>
      <s:element name="group" type="s:string" minOccurs="0" maxOccurs="1"/>
      <s:element minOccurs="1" maxOccurs="1" name="privacy" type="tns:Privacy"/>
      <s:element minOccurs="1" maxOccurs="1" name="isInWorkGroup" type="s:boolean"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**accountName:** A string that specifies the **login name** of the **user profile** to be updated. This value MUST be less than or equal to 400 characters in length. If this element is not specified or does not have a value, the protocol server MUST use the user profile of the **current user**. The protocol server MUST treat **accountName** in a case insensitive manner for retrieval of the user profile.

If **accountName** does not correspond to an existing user profile, the protocol server MUST send a **SOAP fault** message.

**colleagueAccountName:** A string that specifies the login name of the user profile of the **colleague** to be added to the user profile specified by **accountName**. This value MUST be less than or equal to 400 characters in length. If this element is not specified or does not have a value, the protocol server MUST use the user profile of the current user, and return a SOAP fault message if the current user does not have a user profile. The protocol server MUST treat **colleagueAccountName** in a case insensitive manner for retrieval of the user profile.

If **colleagueAccountName** does not correspond to an existing user profile, the protocol server MUST send a SOAP fault message.

The user specified by **colleagueAccountName** MUST NOT be the same as the user specified by **accountName**.

If the user specified by **accountName** and the user specified by **colleagueAccountName** are already colleagues, the protocol server MUST send a SOAP fault message.

If the **accountName** is not specified or does not have a value and the **colleagueAccountName** is not specified or does not have a value, the protocol server MUST send a SOAP fault message.

**group:** A string that refers to a **group (2)**. This value MUST be less than or equal to 50 characters in length.

**privacy:** A value that MUST correspond to a **Privacy** enumeration (section [2.2.5.4](#)). It specifies the set of users who can view this colleague association. This value MUST NOT be "NotSet" as specified in **Privacy** (section 2.2.5.4). If this value is set to "NotSet" the server MUST send a SOAP fault message.

**isInWorkGroup:** A Boolean value which, when **true**, the protocol server MUST allow the user specified in **colleagueAccountName** to see user profile information that falls within the **organization privacy level** for the user specified by **accountName**.

### 3.1.4.2.2 AddColleagueWithoutEmailNotificationResponse

The **AddColleagueWithoutEmailNotificationResponse** element is the body of the response **WSDL** message for the **AddColleagueWithoutEmailNotification** WSDL operation.

```
<s:element name="AddColleagueWithoutEmailNotificationResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="AddColleagueWithoutEmailNotificationResult" type="tns:ContactData"
minOccurs="0"
          maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**AddColleagueWithoutEmailNotificationResult:** A **ContactData** element (section [2.2.4.11](#)) that corresponds to the created **colleague** association.

### 3.1.4.3 AddLink

The **AddLink** **WSDL** operation is used to update the specified **user profile** by adding a **quick link**.

```
<wsdl:operation name="AddLink">
  <wsdl:input message="tns:AddLinkSoapIn"/>
  <wsdl:output message="tns:AddLinkSoapOut"/>
</wsdl:operation>
```

The protocol client sends an **AddLinkSoapIn** request WSDL message and the protocol server responds with an **AddLinkSoapOut** response WSDL message. If the user is not authorized to administer the user profile service and is not updating his or her own user profile, the protocol server MUST send a **SOAP fault** message.

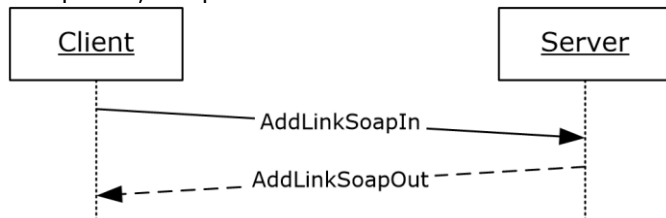


Figure 5: AddLink operation

#### 3.1.4.3.1 Messages

The following **WSDL** message definitions are specific to this operation.

##### 3.1.4.3.1.1 AddLinkSoapIn

The **AddLinkSoapIn** message is the request **WSDL** message for the **AddLink** WSDL operation. The **SOAP action** value of the message is defined as:

```
http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/AddLink
```

The **SOAP body** contains an **AddLink** element.

##### 3.1.4.3.1.2 AddLinkSoapOut

The **AddLinkSoapOut** message is the response **WSDL** message for the **AddLink** WSDL operation.

The **SOAP body** contains an **AddLinkResponse** element.

### 3.1.4.3.2 Elements

The following **XML schema** element definitions are specific to this operation.

#### 3.1.4.3.2.1 AddLink

The **AddLink** element is the body of the request **WSDL** message for the **AddLink** WSDL operation. If there is another **quick link** association in the **user profile** referred to by **accountName** that has the same name, **URL**, and **group (1)** (all treated in a case insensitive manner), the protocol server **MUST** send a **SOAP fault** message.

```
<s:element name="AddLink">
  <s:complexType>
    <s:sequence>
      <s:element name="accountName" type="s:string" minOccurs="0" maxOccurs="1"/>
      <s:element minOccurs="0" maxOccurs="1" name="name" type="s:string"/>
      <s:element minOccurs="0" maxOccurs="1" name="url" type="s:string"/>
      <s:element minOccurs="0" maxOccurs="1" name="group" type="s:string"/>
      <s:element minOccurs="1" maxOccurs="1" name="privacy" type="tns:Privacy"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**accountName:** A string that specifies the **login name** of the user profile to return the elements for. This value **MUST** be less than or equal to 400 characters in length. The protocol server **MUST** treat **accountName** in a case insensitive manner for retrieval of the user profile.

If **accountName** is not specified or does not have a value and there is not an existing user profile for the **current user**, then a new user profile **MUST** be created which represents the current user.

If **accountName** is not specified or does not have a value and there is an existing user profile for the current user, then the protocol server **MUST** return elements for that user profile.

If **accountName** is specified and does not match the login name of an existing user profile, the protocol server **MUST** send a SOAP fault message.

**name:** A string that specifies the display text for the new quick link. This value **MUST** be less than or equal to 50 characters.

**url:** The URL that the quick link refers to. This value **MUST** be less than or equal to 2048 characters and **MUST** be a properly formatted URL as specified in [\[RFC2396\]](#). If the **url** is null or an empty string, the server **MUST** return a SOAP fault message.

**group:** A string that refers to a group (2). This value **MUST** be less than or equal to 50 characters.

**privacy:** This specifies the set of users who can view this quick link. The value **MUST** correspond to a value in the **Privacy** (section [2.2.5.4](#)) enumeration. This value **MUST NOT** be "NotSet".

#### 3.1.4.3.2.2 AddLinkResponse

The **AddLinkResponse** element is the body of the response **WSDL** message for the **AddLink** WSDL operation.

```
<s:element name="AddLinkResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="AddLinkResult" type="tns:QuickLinkData" minOccurs="0"
        maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**AddLinkResult:** A **QuickLinkData** (section [2.2.4.18](#)) element corresponding to the created object.



### 3.1.4.4 AddMembership

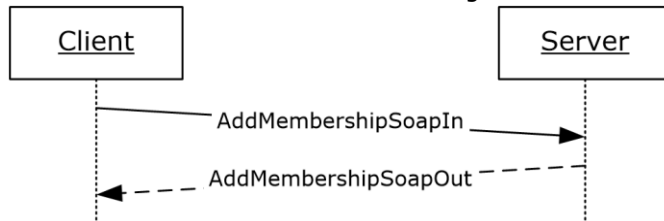
The **AddMembership** **WSDL** operation is used to update the specified **user profile** by adding a **membership**.

```
<wsdl:operation name="AddMembership">
  <wsdl:input message="tns:AddMembershipSoapIn"/>
  <wsdl:output message="tns:AddMembershipSoapOut"/>
</wsdl:operation>
```

The protocol client sends an **AddMembershipSoapIn** request WSDL message and the protocol server responds with an **AddMembershipSoapOut** response WSDL message.

If the user is not authorized to administer the user profile service and is not updating his or her own user profile, the protocol server **MUST** send a **SOAP fault** message.

Additionally, if the user does not have the **Use Personalization Features** permission, the protocol server **MUST** send a SOAP fault message.



**Figure 6: AddMembership operation**

#### 3.1.4.4.1 Messages

The following **WSDL** message definitions are specific to this operation.

##### 3.1.4.4.1.1 AddMembershipSoapIn

The **AddMembershipSoapIn** message is the request **WSDL** message for the **AddMembership** WSDL operation.

The **SOAP action** value of the message is defined as:

```
http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/AddMembership
```

The **SOAP body** contains an **AddMembership** element.

##### 3.1.4.4.1.2 AddMembershipSoapOut

The **AddMembershipSoapOut** message is the response **WSDL** message for the **AddMembership** WSDL operation.

The **SOAP body** contains an **AddMembershipResponse** element.

#### 3.1.4.4.2 Elements

The following **XML schema** element definitions are specific to this operation.

##### 3.1.4.4.2.1 AddMembership

The **AddMembership** element is the body of the request **WSDL** message for the **AddMembership** WSDL operation.

```
<s:element name="AddMembership">
  <s:complexType>
    <s:sequence>
      <s:element name="accountName" type="s:string" minOccurs="0" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

```

    <s:element minOccurs="0" maxOccurs="1" name="membershipInfo"
type="tns:MembershipData"/>
    <s:element minOccurs="0" maxOccurs="1" name="group" type="s:string"/>
    <s:element name="privacy" type="tns:Privacy"/>
  </s:sequence>
</s:complexType>
</s:element>

```

**accountName:** A string that specifies the **login name** of the **user profile** to return the elements for. This value MUST be less than or equal to 400 characters in length. The protocol server MUST treat **accountName** in a case insensitive manner for retrieval of the user profile.

If **accountName** is not specified or does not have a value and there is not an existing user profile for the **current user**, then a new user profile MAY <7> be created which represents the current user.

If **accountName** is not specified or does not have a value and there is an existing user profile for the current user, then the protocol server MUST return elements for that user profile.

If **accountName** is specified and does not match the login name of an existing user profile, the protocol server MUST send a **SOAP fault** message.

**membershipInfo:** Identifies the **member group** of the **membership** that will be added to the user profile. The protocol server MUST only use the **MemberGroup** element of the **MembershipData** (section 2.2.4.13) element and MUST ignore all other elements.

If the **membershipInfo** element specified does not correspond to an existing member group, the protocol server MUST send a SOAP fault message.

If the user profile already contains a membership to this member group, the protocol server MUST send a SOAP fault message.

**group:** A string that refers to a **group (2)**. This value MUST be less than or equal to 50 characters.

**privacy:** A value that MUST correspond to a value in the **Privacy** (section 2.2.5.4) enumeration. This value specifies the set of users who can view this membership. This value MUST NOT be "NotSet".

### 3.1.4.4.2 AddMembershipResponse

The **AddMembershipResponse** element is the body of the response **WSDL** message for the **AddMembership** WSDL operation.

```

<s:element name="AddMembershipResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="AddMembershipResult" type="tns:MembershipData" minOccurs="0"
maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>

```

**AddMembershipResult:** A **MembershipData** (section 2.2.4.13) element corresponding to the created **membership**.

### 3.1.4.5 AddPinnedLink

The **AddPinnedLink WSDL** operation is used to update the specified **user profile** by adding a **pinned link**.

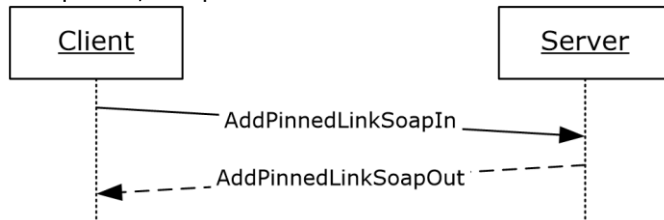
```

<wsdl:operation name="AddPinnedLink">
  <wsdl:input message="tns:AddPinnedLinkSoapIn" />
  <wsdl:output message="tns:AddPinnedLinkSoapOut" />
</wsdl:operation>

```

The protocol client sends an **AddPinnedLinkSoapIn** request WSDL message, and the protocol server responds with an **AddPinnedLinkSoapOut** response WSDL message.

If the user is not authorized to administer the user profile service and is not updating his or her own user profile, the protocol server MUST send a **SOAP fault** message.



**Figure 7: AddPinnedLink operation**

### 3.1.4.5.1 Messages

The following **WSDL** message definitions are specific to this operation.

#### 3.1.4.5.1.1 AddPinnedLinkSoapIn

The **AddPinnedLinkSoapIn** message is the request **WSDL** message for the **AddPinnedLink** WSDL operation.

The **SOAP action** value of the message is defined as:

```
http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/AddPinnedLink
```

The **SOAP body** contains an **AddPinnedLink** element.

#### 3.1.4.5.1.2 AddPinnedLinkSoapOut

The **AddPinnedLinkSoapOut** message is the response **WSDL** message for the **AddPinnedLink** WSDL operation.

The **SOAP body** contains an **AddPinnedLinkResponse** element.

### 3.1.4.5.2 Elements

The following **XML schema** element definitions are specific to this operation.

#### 3.1.4.5.2.1 AddPinnedLink

The **AddPinnedLink** element is the body of the request **WSDL** message for the **AddPinnedLink** WSDL operation.

If a **pinned link** with the same **URL** already exists for the **user profile** associated with **accountName**, the protocol server MUST replace the existing pinned link's name with the **name** specified in this element.

```
<s:element name="AddPinnedLink">
  <s:complexType>
    <s:sequence>
      <s:element name="accountName" type="s:string" minOccurs="0" maxOccurs="1"/>
      <s:element minOccurs="0" maxOccurs="1" name="name" type="s:string"/>
      <s:element minOccurs="0" maxOccurs="1" name="url" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**accountName:** A string that specifies the **login name** of the user profile to be updated. This value MUST be less than or equal to 400 characters in length. If this element is not specified or does not have a value, the protocol server MUST use the user profile of the **current user**, and return a **SOAP**

**fault** message if the current user does not have a user profile. The protocol server MUST treat **accountName** in a case insensitive manner for retrieval of the user profile.

If the **accountName** does not correspond to an existing user profile, the protocol server MUST return a SOAP fault message.

**name:** A string that specifies the display text for the new pinned link. This value MUST be less than or equal to 50 characters.

**url:** The URL of the pinned link. This MUST be less than or equal to 2048 characters and MUST be a properly formatted URL as specified in [\[RFC2396\]](#). If the **url** is null or an empty string, the server MUST return a SOAP fault message.

### 3.1.4.5.2.2 AddPinnedLinkResponse

The **AddPinnedLinkResponse** element is the body of the response **WSDL** message for the **AddPinnedLink** WSDL operation.

```
<s:element name="AddPinnedLinkResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="AddPinnedLinkResult" type="tns:PinnedLinkData" minOccurs="0"
        maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

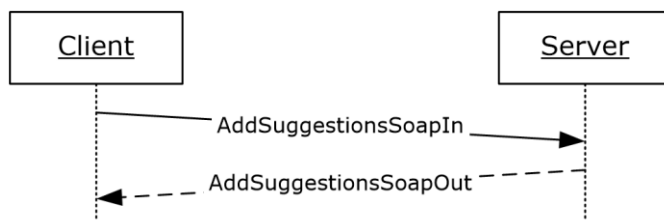
**AddPinnedLinkResult:** A **PinnedLinkData** (section [2.2.4.15](#)) element corresponding to the created **pinned link**.

### 3.1.4.6 AddSuggestions

The **AddSuggestions** **WSDL** operation is used to update the **user profile** of the **current user** with new suggestions of **colleagues** or **keywords**.

```
<wsdl:operation name="AddSuggestions">
  <wsdl:input message="tns:AddSuggestionsSoapIn"/>
  <wsdl:output message="tns:AddSuggestionsSoapOut"/>
</wsdl:operation>
```

The protocol client sends an **AddSuggestionsSoapIn** request WSDL message and the protocol server responds with an **AddSuggestionsSoapOut** response WSDL message.



**Figure 8: AddSuggestions operation**

#### 3.1.4.6.1 Messages

The following **WSDL** message definitions are specific to this operation.

##### 3.1.4.6.1.1 AddSuggestionsSoapIn

The **AddSuggestionsSoapIn** message is the request **WSDL** message for the **AddSuggestions** WSDL operation.

The **SOAP action** value of the message is defined as:

```
http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/AddSuggestions
```

The **SOAP body** contains an **AddSuggestions** element.

### 3.1.4.6.1.2 AddSuggestionsSoapOut

The **AddSuggestionsSoapOut** message is the response **WSDL** message for the **AddSuggestions** WSDL operation.

The **SOAP body** contains an **AddSuggestionsResponse** element.

### 3.1.4.6.2 Elements

The following **XML schema** element definitions are specific to this operation.

#### 3.1.4.6.2.1 AddSuggestions

The **AddSuggestions** element is the body of the request **WSDL** message for the **AddSuggestions** WSDL operation.

```
<s:element name="AddSuggestions">
  <s:complexType>
    <s:sequence><s:element minOccurs="1" maxOccurs="1" name="type"
type="tns:SuggestionType" />
      <s:element minOccurs="0" maxOccurs="1" name="suggestions" type="tns:ArrayOfString"
/>
      <s:element minOccurs="0" maxOccurs="1" name="weights" type="tns:ArrayOfDouble" />
    </s:sequence>
  </s:complexType>
</s:element>
```

**SuggestionType**: A **SuggestionType** (section [2.2.5.5](#)) element that specifies the type of suggestions to be added to the **user profile** of the **current user**.

**suggestions**: An array of strings containing colleagues or **keywords**.

**weights**: An array of double values. **suggestions** are sorted in descending **weights** values.

**suggestions** and **weights** MUST be paired.

If **SuggestionType** is Colleague, **suggestions** MUST be either **accountName** or **email**. If **SuggestionType** is Keyword, **suggestions** MUST be keyword.

#### 3.1.4.6.2.2 AddSuggestionsResponse

The **AddSuggestionsResponse** element is the body of the response **WSDL** message for the **AddSuggestions** WSDL operation.

```
<s:element name="AddSuggestionsResponse">
  <s:complexType/>
</s:element>
```

### 3.1.4.7 CreateMemberGroup

The **CreateMemberGroup WSDL** operation is used to create a new **member group**.

```
<wsdl:operation name="CreateMemberGroup">
  <wsdl:input message="tns:CreateMemberGroupSoapIn"/>
  <wsdl:output message="tns:CreateMemberGroupSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **CreateMemberGroupSoapIn** request WSDL message and the protocol server responds with a **CreateMemberGroupSoapOut** response WSDL message. If the user is not authorized to administer the user profile service, the protocol server MUST send a **SOAP fault** message.

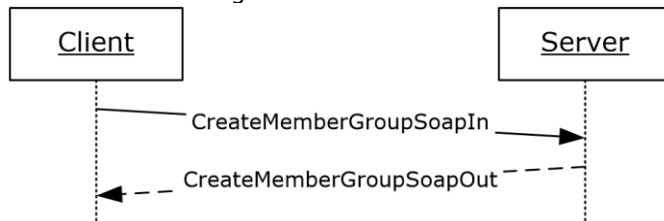


Figure 9: CreateMemberGroup operation

#### 3.1.4.7.1 Messages

The following **WSDL** message definitions are specific to this operation.

##### 3.1.4.7.1.1 CreateMemberGroupSoapIn

The **CreateMemberGroupSoapIn** message is the request **WSDL** message for the **CreateMemberGroup** WSDL operation.

The **SOAP action** value of the message is defined as:

```
http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/CreateMemberGroup
```

The **SOAP body** contains a **CreateMemberGroup** element.

##### 3.1.4.7.1.2 CreateMemberGroupSoapOut

The **CreateMemberGroupSoapOut** message is the response **WSDL** message for the **CreateMemberGroup** WSDL operation.

The **SOAP body** contains a **CreateMemberGroupResponse** element.

#### 3.1.4.7.2 Elements

The following **XML schema** element definitions are specific to this operation.

##### 3.1.4.7.2.1 CreateMemberGroup

The **CreateMemberGroup** element is the body of the request **WSDL** message for the **CreateMemberGroup** WSDL operation.

```
<s:element name="CreateMemberGroup">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="membershipInfo"
type="tns:MembershipData"/>
    </s:sequence>
  </s:complexType>
```

```
</s:element>
```

**membershipInfo:** The information for the **member group** to be created. The protocol server MUST only use the **MemberGroup**, **DisplayName**, **MailNickname**, and **Url** elements of the **MembershipData** (section 2.2.4.13) element and MUST ignore all other elements. If **DisplayName** or **MailNickname** are null the server MUST send a SOAP fault message.

If the **MemberGroupData** (section 2.2.4.12) element of **membershipInfo** element corresponds to an existing member group on the protocol server, the protocol server MUST send a **SOAP fault** message.

### 3.1.4.7.2.2 CreateMemberGroupResponse

The **CreateMemberGroupResponse** element is the body of the response **WSDL** message for the **CreateMemberGroup** WSDL operation.

```
<s:element name="CreateMemberGroupResponse">
  <s:complexType/>
</s:element>
```

### 3.1.4.8 CreateUserProfileByAccountName

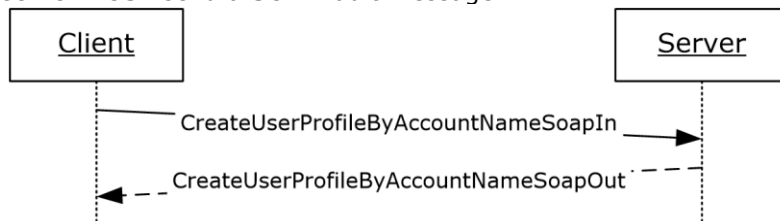
The **CreateUserProfileByAccountName** **WSDL** operation is used to create a **user profile** for the specified user **login name**.

```
<wsdl:operation name="CreateUserProfileByAccountName">
  <wsdl:input message="tns:CreateUserProfileByAccountNameSoapIn"/>
  <wsdl:output message="tns:CreateUserProfileByAccountNameSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **CreateUserProfileByAccountNameSoapIn** request WSDL message and the protocol server responds with a **CreateUserProfileByAccountNameSoapOut** response WSDL message.

If the user is not authorized to administer the user profile service and is not creating his or her own user profile, the protocol server MUST send a **SOAP fault** message.

Additionally, if the user does not have the **Use Personalization Features** permission, the protocol server MUST send a SOAP fault message.



**Figure 10: CreateUserProfileByAccountName operation**

#### 3.1.4.8.1 Messages

The following **WSDL** message definitions are specific to this operation.

##### 3.1.4.8.1.1 CreateUserProfileByAccountNameSoapIn

The **CreateUserProfileByAccountNameSoapIn** message is the request **WSDL** message for the **CreateUserProfileByAccountName** WSDL operation.

The **SOAP action** value of the message is defined as:

<http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/CreateUserProfileByAccountName>

The **SOAP body** contains a **CreateUserProfileByAccountName** element.

#### 3.1.4.8.1.2 CreateUserProfileByAccountNameSoapOut

The **CreateUserProfileByAccountNameSoapOut** message is the response **WSDL** message for the **CreateUserProfileByAccountName** WSDL operation.

The **SOAP body** contains a **CreateUserProfileByAccountNameResponse** element.

#### 3.1.4.8.2 Elements

The following **XML schema** element definitions are specific to this operation.

##### 3.1.4.8.2.1 CreateUserProfileByAccountName

The **CreateUserProfileByAccountName** element is the body of the request **WSDL** message for the **CreateUserProfileByAccountName** WSDL operation.

```
<s:element name="CreateUserProfileByAccountName">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="accountName" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**accountName:** A string that specifies the **login name** of the **user profile** to be created. This value MUST be less than or equal to 400 characters in length and MUST be specified. The protocol server MUST treat **accountName** in a case insensitive manner for retrieval of the user profile.

If a user profile with the specified login name already exists, the protocol server MUST send a **SOAP fault** message.

If the login name is not found in the **directory service (DS)**, the protocol server MUST send a SOAP fault message.

##### 3.1.4.8.2.2 CreateUserProfileByAccountNameResponse

The **CreateUserProfileByAccountNameResponse** element is the body of the response **WSDL** message for the **CreateUserProfileByAccountName** WSDL operation.

```
<s:element name="CreateUserProfileByAccountNameResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="CreateUserProfileByAccountNameResult"
type="tns:ArrayOfPropertyData" />
    </s:sequence>
  </s:complexType>
</s:element>
```

**CreateUserProfileByAccountNameResult:** An **ArrayOfPropertyData** (section [2.2.4.6](#)) element containing all property data for the newly created **user profile**.

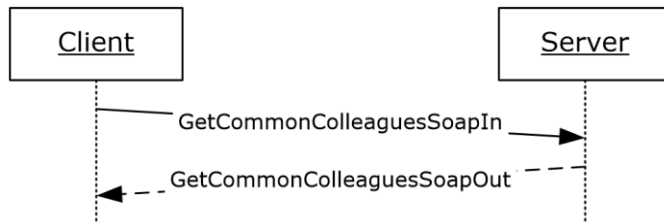


### 3.1.4.9 GetCommonColleagues

The **GetCommonColleagues WSDL** operation is used to get the colleagues that are common to the specified **user profile** and the user profile of the **current user**.

```
<wsdl:operation name="GetCommonColleagues">
  <wsdl:input message="tns:GetCommonColleaguesSoapIn"/>
  <wsdl:output message="tns:GetCommonColleaguesSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **GetCommonColleaguesSoapIn** request WSDL message and the protocol server responds with a **GetCommonColleaguesSoapOut** response WSDL message.



**Figure 11: GetCommonColleagues operation**

#### 3.1.4.9.1 Messages

The following **WSDL** message definitions are specific to this operation.

##### 3.1.4.9.1.1 GetCommonColleaguesSoapIn

The **GetCommonColleaguesSoapIn** message is the request **WSDL** message for the **GetCommonColleagues** WSDL operation.

The **SOAP action** value of the message is defined as:

```
http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetCommonColleagues
```

The **SOAP body** contains a **GetCommonColleagues** element.

##### 3.1.4.9.1.2 GetCommonColleaguesSoapOut

The **GetCommonColleaguesSoapOut** message is the response **WSDL** message for the **GetCommonColleagues** WSDL operation.

The **SOAP body** contains a **GetCommonColleaguesResponse** element.

#### 3.1.4.9.2 Elements

The following **XML schema** element definitions are specific to this operation.

##### 3.1.4.9.2.1 GetCommonColleagues

The **GetCommonColleagues** element is the body of the request **WSDL** message for the **GetCommonColleagues** WSDL operation.

```
<s:element name="GetCommonColleagues">
  <s:complexType>
```

```

    <s:sequence>
      <s:element name="accountName" type="s:string" minOccurs="0" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>

```

**accountName:** A string that specifies the **login name** of the **user profile** to be compared with. This value MUST be less than or equal to 400 characters in length. If this value matches the login name of the **current user**, the protocol server MUST return the colleagues for the user profile of the current user. If this element is not specified or does not have a value, AND a user profile for the current user exists, the protocol server MUST return the colleagues for the user profile of the current user. If this element is not specified or does not have a value, AND a user profile for the current user does not exist, the protocol server MUST create a user profile for the current user and return the colleagues for the user profile of the current user.. The protocol server MUST treat **accountName** in a case insensitive manner for retrieval of the user profile.

If the **accountName** does not correspond to an existing user profile, the protocol server MUST send a **SOAP fault** message.

### 3.1.4.9.2.2 GetCommonColleaguesResponse

The **GetCommonColleaguesResponse** element is the body of the response **WSDL** message for the **GetCommonColleagues** WSDL operation.

```

<s:element name="GetCommonColleaguesResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="GetCommonColleaguesResult" type="tns:ArrayOfContactData"
        minOccurs="0" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>

```

**GetCommonColleaguesResult:** A set of **ContactData** (section [2.2.4.11](#)) elements corresponding to the common colleagues. The **Privacy** element of **ContactData** MUST be returned as "NotSet", the Group MUST be NULL, the **ID** MUST be zero, and **IsInWorkGroup** MUST be **false**.

The values returned MUST be filtered according to the **privacy level**.

### 3.1.4.10 GetCommonManager

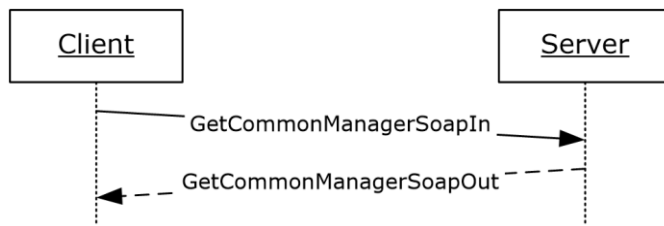
The **GetCommonManager** **WSDL** operation is used to get the lowest level manager in common between the specified **user profile** and the user profile of the **current user**.

```

<wsdl:operation name="GetCommonManager">
  <wsdl:input message="tns:GetCommonManagerSoapIn"/>
  <wsdl:output message="tns:GetCommonManagerSoapOut"/>
</wsdl:operation>

```

The protocol client sends a **GetCommonManagerSoapIn** request WSDL message and the protocol server responds with a **GetCommonManagerSoapOut** response WSDL message.



**Figure 12: GetCommonManager operation**

### 3.1.4.10.1 Messages

The following **WSDL** message definitions are specific to this operation.

#### 3.1.4.10.1.1 GetCommonManagerSoapIn

The **GetCommonManagerSoapIn** message is the request **WSDL** message for the **GetCommonManager** WSDL operation.

The **SOAP action** value of the message is defined as:

```
http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetCommonManager
```

The **SOAP body** contains a **GetCommonManager** element.

#### 3.1.4.10.1.2 GetCommonManagerSoapOut

The **GetCommonManagerSoapOut** message is the response **WSDL** message for the **GetCommonManager** WSDL operation.

The **SOAP body** contains a **GetCommonManagerResponse** element.

### 3.1.4.10.2 Elements

The following **XML schema** element definitions are specific to this operation.

#### 3.1.4.10.2.1 GetCommonManager

The **GetCommonManager** element is the body of the request **WSDL** message for the **GetCommonManager** WSDL operation.

```

<s:element name="GetCommonManager">
  <s:complexType>
    <s:sequence>
      <s:element name="accountName" type="s:string" minOccurs="0" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
  
```

**accountName:** A string that specifies the **login name** of the **user profile** to be compared with. This value **MUST** be less than or equal to 400 characters in length. If this element is not specified or does not have a value, the protocol server **MUST** use the user profile of the **current user**, and return a **SOAP fault** message if the current user does not have a user profile. The protocol server **MUST** treat **accountName** in a case insensitive manner for retrieval of the user profile.

If the **accountName** does not correspond to an existing user profile, the protocol server **MUST** send a SOAP fault message.

### 3.1.4.10.2.2 GetCommonManagerResponse

The **GetCommonManagerResponse** element is the body of the response **WSDL** message for the **GetCommonManager** WSDL operation.

```
<s:element name="GetCommonManagerResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="GetCommonManagerResult" type="tns:ContactData" minOccurs="0"
        maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**GetCommonManagerResult:** A **ContactData** (section 2.2.4.11) element corresponding to the common **Manager** property. The user specified by this element **MUST NOT** be the same user as the **current user** or as the user specified by **AccountName**.

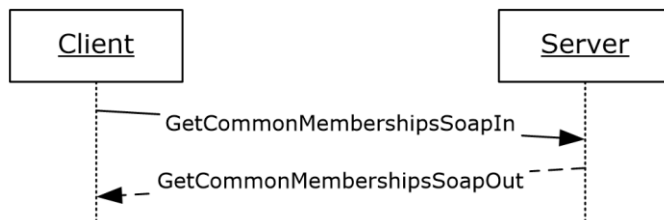
If no common Manager can be found, the protocol server **MUST** send a SOAP fault message.

### 3.1.4.11 GetCommonMemberships

The **GetCommonMemberships** **WSDL** operation is used to get all of the **membership** properties in common between the specified **user profile** and the user profile of the **current user**.

```
<wsdl:operation name="GetCommonMemberships">
  <wsdl:input message="tns:GetCommonMembershipsSoapIn"/>
  <wsdl:output message="tns:GetCommonMembershipsSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **GetCommonMembershipsSoapIn** request WSDL message and the protocol server responds with a **GetCommonMembershipsSoapOut** response WSDL message.



**Figure 13: GetCommonMemberships operation**

#### 3.1.4.11.1 Messages

The following **WSDL** message definitions are specific to this operation.

##### 3.1.4.11.1.1 GetCommonMembershipsSoapIn

The **GetCommonMembershipsSoapIn** message is the request **WSDL** message for the **GetCommonMemberships** WSDL operation.

The **SOAP action** value of the message is defined as:

```
http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetCommonMemberships
```

The **SOAP body** contains a **GetCommonMemberships** element.

#### 3.1.4.11.1.2 GetCommonMembershipsSoapOut

The **GetCommonMembershipsSoapOut** message is the response **WSDL** message for the **GetCommonMemberships** WSDL operation.

The **SOAP body** contains a **GetCommonMembershipsResponse** element.

#### 3.1.4.11.2 Elements

The following **XML schema** element definitions are specific to this operation.

##### 3.1.4.11.2.1 GetCommonMemberships

The **GetCommonMemberships** element is the body of the request **WSDL** message for the **GetCommonMemberships** WSDL operation.

```
<s:element name="GetCommonMemberships">
  <s:complexType>
    <s:sequence>
      <s:element name="accountName" type="s:string" minOccurs="0" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**accountName:** A string that specifies the **login name** of the **user profile** to be compared with. This value MUST be less than or equal to 400 characters in length. If this value matches the login name of the **current user**, the protocol server MUST return the memberships associated with user profile of the current user. If this element is not specified or does not have a value, AND a user profile for the current user exists, the protocol server MUST return the memberships associated with the user profile of the current user. If this element is not specified or does not have a value, AND a user profile for the current user does not exist, the protocol server MAY [≤8>](#) create a user profile for the current user and return the memberships associated with the user profile of the current user. The protocol server MUST treat **accountName** in a case insensitive manner for retrieval of the user profile.

If the **accountName** does not correspond to an existing user profile, the protocol server MUST send a **SOAP fault** message.

##### 3.1.4.11.2.2 GetCommonMembershipsResponse

The **GetCommonMembershipsResponse** element is the body of the response **WSDL** message for the **GetCommonMemberships** WSDL operation.

```
<s:element name="GetCommonMembershipsResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="GetCommonMembershipsResult" type="tns:ArrayOfMembershipData"
        minOccurs="0" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**GetCommonMembershipsResult:** A set of **MembershipData** (section [2.2.4.13](#)) elements corresponding to the **memberships** that the users have in common. The **Privacy** element of **MembershipData** MUST be returned as "NotSet", the Group MUST be NULL and the **ID** MUST be "0".

The values returned MUST be filtered according to the privacy enumeration.

### 3.1.4.12 GetInCommon

The **GetInCommon WSDL** operation is used to get all common colleagues, all common **memberships**, and the lowest level common **Manager** property between the specified **user profile** and the user profile of the **current user**.

```
<wsdl:operation name="GetInCommon">
  <wsdl:input message="tns:GetInCommonSoapIn"/>
  <wsdl:output message="tns:GetInCommonSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **GetInCommonSoapIn** request WSDL message and the protocol server responds with a **GetInCommonSoapOut** response WSDL message.

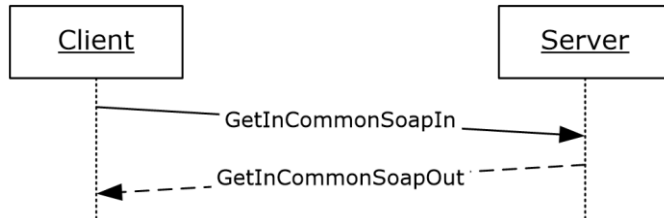


Figure 14: GetInCommon operation

#### 3.1.4.12.1 Messages

The following **WSDL** message definitions are specific to this operation.

##### 3.1.4.12.1.1 GetInCommonSoapIn

The **GetInCommonSoapIn** message is the request **WSDL** message for the **GetInCommon** WSDL operation.

The **SOAP action** value of the message is defined as:

```
http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetInCommon
```

The **SOAP body** contains a **GetInCommon** element.

##### 3.1.4.12.1.2 GetInCommonSoapOut

The **GetInCommonSoapOut** message is the response **WSDL** message for the **GetInCommon** WSDL operation.

The **SOAP body** contains a **GetInCommonResponse** element.

#### 3.1.4.12.2 Elements

The following **XML schema** element definitions are specific to this operation.

##### 3.1.4.12.2.1 GetInCommon

The **GetInCommon** element is the body of the request **WSDL** message for the **GetInCommon** WSDL operation.

```
<s:element name="GetInCommon">
  <s:complexType>
```

```

    <s:sequence>
      <s:element name="accountName" type="s:string" minOccurs="0" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>

```

**accountName:** A string that specifies the **login name** of the **user profile** to be compared with. This value MUST be less than or equal to 400 characters in length. The protocol server MUST treat **accountName** in a case insensitive manner for retrieval of the user profile.

If **accountName** is not specified or does not have a value and there is not an existing user profile for the **current user**, then a new user profile MUST be created which represents the current user.

If **accountName** is not specified or does not have a value and there is an existing user profile for the current user, then the protocol server MUST return elements for ONLY that user profile.

If **accountName** is specified and does not match the login name of an existing user profile, the protocol server MUST send a **SOAP fault** message.

### 3.1.4.12.2.2 GetInCommonResponse

The **GetInCommonResponse** element is the body of the response **WSDL** message for the **GetInCommon** WSDL operation.

```

<s:element name="GetInCommonResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="GetInCommonResult" type="tns:InCommonData" minOccurs="0"
        maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>

```

**GetInCommonResult:** An **InCommonData** (section [3.1.4.12.3.1](#)) element containing the colleague properties, **membership** properties, and manager property in common between the **user profile** of the **current user** and the specified user profile.

The values returned MUST be filtered according to the **privacy level**.

If no common Manager can be found, the protocol server MUST either send a SOAP fault or return ContactData with the Manager field set to null. [<9>](#)

### 3.1.4.12.3 Complex Types

The following **XML schema** complex type definitions are specific to this operation.

#### 3.1.4.12.3.1 InCommonData

The **InCommonData** complex type is used to represent the returned common manager, colleagues, and **memberships**.

```

<s:complexType name="InCommonData">
  <s:sequence>
    <s:element name="Manager" type="tns:ContactData" minOccurs="0" maxOccurs="1"/>
    <s:element name="Colleagues" type="tns:ArrayOfContactData" minOccurs="0"
      maxOccurs="1"/>
    <s:element name="Memberships" type="tns:ArrayOfMembershipData" minOccurs="0"
      maxOccurs="1"/>
  </s:sequence>
</s:complexType>

```

**Manager:** A **ContactData** (section [2.2.4.11](#)) element corresponding to the common manager.

**Colleagues:** A set of **ContactData** (section [2.2.4.11](#)) elements corresponding to the common colleague. The same Privacy element restrictions that apply to the **GetCommonColleaguesResponse** type (section [3.1.4.9.2.2](#)) also apply to this data set.

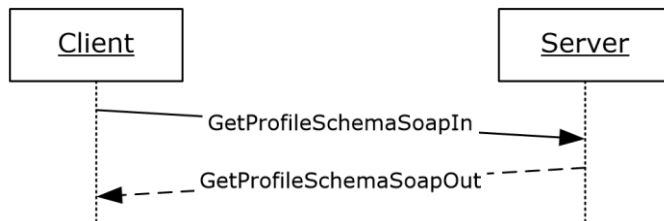
**Memberships:** A set of **MembershipData** (section [2.2.4.13](#)) elements corresponding to the common memberships.

### 3.1.4.13 GetProfileSchema

The **GetProfileSchema** **WSDL** operation is used to retrieve the list of properties associated with a particular user profile schema.

```
<wsdl:operation name="GetProfileSchema">
  <wsdl:input message="tns:GetProfileSchemaSoapIn" />
  <wsdl:output message="tns:GetProfileSchemaSoapOut" />
</wsdl:operation>
```

The protocol client sends a **GetProfileSchemaSoapIn** request WSDL message and the protocol server responds with a **GetProfileSchemaSoapOut** response WSDL message.



**Figure 15: GetProfileSchema operation**

#### 3.1.4.13.1 Messages

The following **WSDL** message definitions are specific to this operation.

##### 3.1.4.13.1.1 GetProfileSchemaSoapIn

The **GetProfileSchemaSoapIn** message is the request **WSDL** message for the **GetProfileSchema** WSDL operation.

The **SOAP action** value of the message is defined as:

```
http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetProfileSchema
```

The **SOAP body** contains a **GetProfileSchema** element.

##### 3.1.4.13.1.2 GetProfileSchemaSoapOut

The **GetProfileSchemaSoapOut** message is the response **WSDL** message for the **GetProfileSchema** WSDL operation.

The **SOAP body** contains a **GetProfileSchemaResponse** element.

#### 3.1.4.13.2 Elements

The following **XML schema** element definitions are specific to this operation.



### 3.1.4.13.2.1 GetProfileSchema

The **GetProfileSchema** element is the body of the request **WSDL** message for the **GetProfileSchema** WSDL operation.

```
<s:element name="GetProfileSchema">
  <s:complexType>
    <s:sequence>
      <s:element name="schemaName" type="s:string" minOccurs="0" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**schemaName:** A string that specifies the profile schema name to retrieve the list of properties on. This value **MUST** be less than or equal to 400 characters in length. If this element is not specified or does not have a value, the protocol server **MUST** send a **SOAP fault** message. The protocol server **MUST** treat **schemaName** in a case insensitive manner for retrieval of the **user profile**.

If the **schemaName** does not correspond to an existing schema, the protocol server **MUST** send a SOAP fault message.

### 3.1.4.13.2.2 GetProfileSchemaResponse

The **GetProfileSchemaResponse** element is the body of the response **WSDL** message for the **GetProfileSchema** WSDL operation.

```
<s:element name="GetProfileSchemaResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="GetProfileSchemaResult"
type="tns:ArrayOfPropertyInfo"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

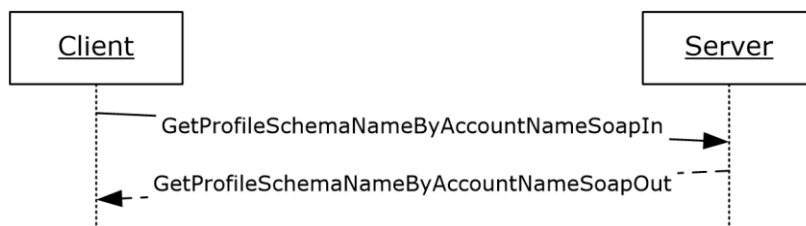
**GetProfileSchemaResult:** The protocol server **MUST** return a set of **PropertyInfo** (section [2.2.4.17](#)) items in the response WSDL message.

### 3.1.4.14 GetProfileSchemaNameByAccountName

The **GetProfileSchemaNameByAccountName** **WSDL** operation is used to retrieve the user profile schema name corresponding to the specified user.

```
<wsdl:operation name="GetProfileSchemaNameByAccountName">
  <wsdl:input message="tns:GetProfileSchemaNameByAccountNameSoapIn"/>
  <wsdl:output message="tns:GetProfileSchemaNameByAccountNameSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **GetProfileSchemaNameByAccountNameSoapIn** request WSDL message and the protocol server responds with a **GetProfileSchemaNameByAccountNameSoapOut** response WSDL message.



**Figure 16: GetProfileSchemaNameByAccountName operation**

### 3.1.4.14.1 Messages

The following **WSDL** message definitions are specific to this operation.

#### 3.1.4.14.1.1 GetProfileSchemaNameByAccountNameSoapIn

The **GetProfileSchemaNameByAccountNameSoapIn** message is the request **WSDL** message for the **GetProfileSchemaNameByAccountName** WSDL operation.

The **SOAP action** value of the message is defined as:

```
http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetProfileSchemaNameByAccountName
```

The **SOAP body** contains a **GetProfileSchemaNameByAccountName** element.

#### 3.1.4.14.1.2 GetProfileSchemaNameByAccountNameSoapOut

The **GetProfileSchemaNameByAccountNameSoapOut** message is the response **WSDL** message for the **GetProfileSchemaNameByAccountName** WSDL operation.

The **SOAP body** contains a **GetProfileSchemaNameByAccountNameResponse** element.

### 3.1.4.14.2 Elements

The following **XML schema** element definitions are specific to this operation.

#### 3.1.4.14.2.1 GetProfileSchemaNameByAccountName

The **GetProfileSchemaNameByAccountName** element is the body of the request **WSDL** message for the **GetProfileSchemaNameByAccountName** WSDL operation.

```

<s:element name="GetProfileSchemaNameByAccountName">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="accountName" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>
  
```

**accountName:** A string that specifies the **login name** of the **user profile** for which to return the name of the profile schema. This value **MUST** be less than or equal to 400 characters in length. The protocol server **MUST** treat **accountName** in a case insensitive manner for retrieval of the user profile.

If **accountName** is not specified or does not have a value and there is not an existing user profile for the **current user**, then a new user profile **MUST** be created which represents the current user and the protocol server **MUST** return the name of the profile schema corresponding to that user profile.

If **accountName** is not specified or does not have a value and there is an existing user profile for the current user, then the protocol server MUST return the name of the profile schema corresponding to the user profile of the current user.

If **accountName** is specified and does not match the login name of an existing user profile, the protocol server MUST send a **SOAP fault** message.

#### 3.1.4.14.2.2 GetProfileSchemaNameByAccountNameResponse

The **GetProfileSchemaNameByAccountNameResponse** element is the body of the response **WSDL** message for the **GetProfileSchemaNameByAccountName** WSDL operation.

```
<s:element name="GetProfileSchemaNameByAccountNameResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1"
        name="GetProfileSchemaNameByAccountNameResult" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**GetProfileSchemaNameByAccountNameResult:** A string that represents the name of the profile schema for the specified user.

#### 3.1.4.15 GetProfileSchemaNames

The **GetProfileSchemaNames** **WSDL** operation is used to retrieve a list of profile schema names in the **user profile**.

```
<wsdl:operation name="GetProfileSchemaNames">
  <wsdl:input message="tns:GetProfileSchemaNamesSoapIn"/>
  <wsdl:output message="tns:GetProfileSchemaNamesSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **GetProfileSchemaNamesSoapIn** request WSDL message and the protocol server responds with a **GetProfileSchemaNamesSoapOut** response WSDL message.

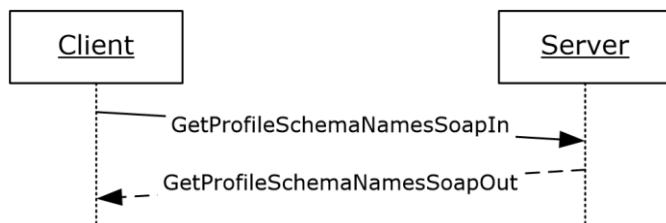


Figure 17: GetProfileSchemaNames operation

#### 3.1.4.15.1 Messages

The following **WSDL** message definitions are specific to this operation.

##### 3.1.4.15.1.1 GetProfileSchemaNamesSoapIn

The **GetProfileSchemaNamesSoapIn** message is the request **WSDL** message for the **GetProfileSchemaNames** WSDL operation.

The **SOAP action** value of the message is defined as:

`http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetProfileSchemaNames`

The **SOAP body** contains a **GetProfileSchemaNames** element.

#### 3.1.4.15.1.2 GetProfileSchemaNamesSoapOut

The **GetProfileSchemaNamesSoapOut** message is the response **WSDL** message for the **GetProfileSchemaNames** WSDL operation.

The **SOAP body** contains a **GetProfileSchemaNamesResponse** element.

#### 3.1.4.15.2 Elements

The following **XML schema** element definitions are specific to this operation.

##### 3.1.4.15.2.1 GetProfileSchemaNames

The **GetProfileSchemaNames** element is the body of the request **WSDL** message for the **GetProfileSchemaNames** WSDL operation.

```
<s:element name="GetProfileSchemaNames">
  <s:complexType/>
</s:element>
```

##### 3.1.4.15.2.2 GetProfileSchemaNamesResponse

The **GetProfileSchemaNamesResponse** element is the body of the response **WSDL** message for the **GetProfileSchemaNames** WSDL operation.

```
<s:element name="GetProfileSchemaNamesResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="GetProfileSchemaNamesResult"
        type="tns:ArrayOfString"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

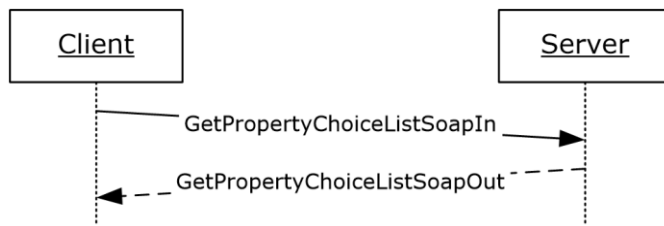
**GetProfileSchemaNamesResult:** An **ArrayOfString** (section [2.2.4.9](#)) type. The protocol server MUST return the set of profile schema names.

#### 3.1.4.16 GetPropertyChoiceList

The **GetPropertyChoiceList** **WSDL** operation is used to retrieve a list of possible values for a property.

```
<wsdl:operation name="GetPropertyChoiceList">
  <wsdl:input message="tns:GetPropertyChoiceListSoapIn"/>
  <wsdl:output message="tns:GetPropertyChoiceListSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **GetPropertyChoiceListSoapIn** request WSDL message and the protocol server responds with a **GetPropertyChoiceListSoapOut** response WSDL message.



**Figure 18: GetPropertyChoiceList operation**

### 3.1.4.16.1 Messages

The following **WSDL** message definitions are specific to this operation.

#### 3.1.4.16.1.1 GetPropertyChoiceListSoapIn

The **GetPropertyChoiceListSoapIn** message is the request **WSDL** message for **GetPropertyChoiceList**.

The **SOAP action** value of the message is defined as:

```
http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetPropertyChoiceList
```

The **SOAP body** contains a **GetPropertyChoiceList** element.

#### 3.1.4.16.1.2 GetPropertyChoiceListSoapOut

The **GetPropertyChoiceListSoapOut** message is the response **WSDL** message for **GetPropertyChoiceList**.

The **SOAP body** contains a **GetPropertyChoiceListResponse** element.

### 3.1.4.16.2 Elements

The following **XML schema** element definitions are specific to this operation.

#### 3.1.4.16.2.1 GetPropertyChoiceList

The **GetPropertyChoiceList** element is the body of the request **WSDL** message for the **GetPropertyChoiceList** WSDL operation.

```

<s:element name="GetPropertyChoiceList">
  <s:complexType>
    <s:sequence>
      <s:element name="propertyName" minOccurs="0" maxOccurs="1" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>
  
```

**propertyName:** A string that specifies the name of the property to get the list of choices for. The protocol server **MUST** trim excess whitespaces from **propertyName** before retrieving the list of choices. The protocol server **MUST** treat **propertyName** in a case insensitive manner for retrieval of the list.

If **propertyName** does not match an existing property on the server, the protocol server **MUST** return a **SOAP fault**. If the property is not available through web service [<10>](#), the protocol server **MUST**

return a **SOAP fault**. If the property does not have a choice list, the protocol server MUST return a **SOAP fault**.

### 3.1.4.16.2.2 GetPropertyChoiceListResponse

The **GetPropertyChoiceListResponse** element is the body of the response **WSDL** message for the **GetPropertyChoiceList** WSDL operation.

```
<s:element name="GetPropertyChoiceListResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="GetPropertyChoiceListResult" type="tns:ArrayOfString" minOccurs="0"
maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**GetPropertyChoiceListResult:** An **ArrayOfString** (section [2.2.4.9](#)) type. The protocol server MUST return the set of values for the given property.

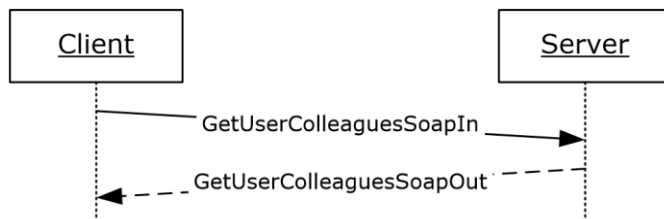
If the property specified by **propertyName** does not have a list of choices, the protocol server MUST return a **SOAP fault**.

### 3.1.4.17 GetUserColleagues

The **GetUserColleagues** **WSDL** operation is used to get all the colleague properties associated with the specified **user profile**.

```
<wsdl:operation name="GetUserColleagues">
  <wsdl:input message="tns:GetUserColleaguesSoapIn"/>
  <wsdl:output message="tns:GetUserColleaguesSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **GetUserColleaguesSoapIn** request WSDL message and the protocol server responds with a **GetUserColleaguesSoapOut** response WSDL message.



**Figure 19: GetUserColleagues operation**

#### 3.1.4.17.1 Messages

The following **WSDL** message definitions are specific to this operation.

##### 3.1.4.17.1.1 GetUserColleaguesSoapIn

The **GetUserColleaguesSoapIn** message is the request **WSDL** message for the **GetUserColleagues** WSDL operation.

The **SOAP action** value of the message is defined as:

```
http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetUserColleagues
```

The **SOAP body** contains a **GetUserColleagues** element.

#### 3.1.4.17.1.2 GetUserColleaguesSoapOut

The **GetUserColleaguesSoapOut** message is the response **WSDL** message for the **GetUserColleagues** WSDL operation.

The **SOAP body** contains a **GetUserColleaguesResponse** element.

#### 3.1.4.17.2 Elements

The following **XML schema** element definitions are specific to this operation.

##### 3.1.4.17.2.1 GetUserColleagues

The **GetUserColleagues** element is the body of the request **WSDL** message for the **GetUserColleagues** WSDL operation.

```
<s:element name="GetUserColleagues">
  <s:complexType>
    <s:sequence>
      <s:element name="accountName" type="s:string" minOccurs="0" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**accountName:** A string that specifies the **login name** of the **user profile** to return the elements for. This value **MUST** be less than or equal to 400 characters in length. The protocol server **MUST** treat **accountName** in a case insensitive manner for retrieval of the user profile.

If **accountName** is not specified or does not have a value and there is not an existing user profile for the **current user**, then a new user profile **MUST** be created which represents the current user and the protocol server **MUST** send a response that includes zero **ContactData** elements.

If **accountName** is not specified or does not have a value and there is an existing user profile for the current user, then the protocol server **MUST** return elements for that user profile.

If **accountName** is specified and does not match the login name of an existing user profile, the protocol server **MUST** send a **SOAP fault** message

##### 3.1.4.17.2.2 GetUserColleaguesResponse

The **GetUserColleaguesResponse** element is the body of the response **WSDL** message for the **GetUserColleagues** WSDL operation.

```
<s:element name="GetUserColleaguesResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="GetUserColleaguesResult" type="tns:ArrayOfContactData"
        minOccurs="0" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**GetUserColleaguesResult:** A set of **ContactData** (section [2.2.4.11](#)) elements corresponding to the colleagues of the specified user.

If the **current user** is not the same as the user specified by **accountName**, and the current user is not authorized to administer the user profile service, the server **MUST** return a **SOAP fault** message.

The protocol server MUST only return **ContactData** elements for **colleagues** the current user has permission to view as specified in section [2.2.5.4](#). The sole exception to this are SharePoint administrator(s) with user personalization permissions; they are permitted to access the data.

If the user specified by **accountName** does not have any colleagues, the response MUST contain zero **ContactData** elements.

### 3.1.4.18 GetUserLinks

The **GetUserLinks** WSDL operation is used to get the **quick link** properties associated with the specified **user profile**.

```
<wsdl:operation name="GetUserLinks">
  <wsdl:input message="tns:GetUserLinksSoapIn"/>
  <wsdl:output message="tns:GetUserLinksSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **GetUserLinksSoapIn** request **WSDL** message and the protocol server responds with a **GetUserLinksSoapOut** response WSDL message.

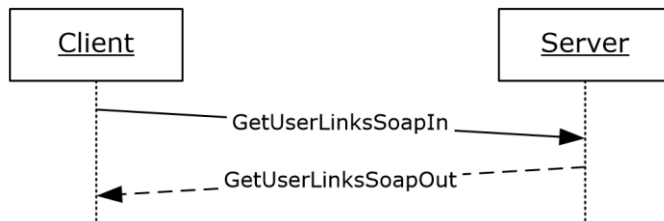


Figure 20: GetUserLinks operation

#### 3.1.4.18.1 Messages

The following **WSDL** message definitions are specific to this operation.

##### 3.1.4.18.1.1 GetUserLinksSoapIn

The **GetUserLinksSoapIn** message is the request **WSDL** message for the **GetUserLinks** WSDL operation.

The **SOAP action** value of the message is defined as:

```
http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetUserLinks
```

The **SOAP body** contains a **GetUserLinks** element.

##### 3.1.4.18.1.2 GetUserLinksSoapOut

The **GetUserLinksSoapOut** message is the response **WSDL** message for the **GetUserLinks** WSDL operation.

The **SOAP body** contains a **GetUserLinksResponse** element.

#### 3.1.4.18.2 Elements

The following **XML schema** element definitions are specific to this operation.

##### 3.1.4.18.2.1 GetUserLinks



The **GetUserLinks** element is the body of the request **WSDL** message for the **GetUserLinks** WSDL operation.

```
<s:element name="GetUserLinks">
  <s:complexType>
    <s:sequence>
      <s:element name="accountName" type="s:string" minOccurs="0" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**accountName:** A string that specifies the **login name** of the **user profile** to return the elements for. This value **MUST** be less than or equal to 400 characters in length. The protocol server **MUST** treat **accountName** in a case insensitive manner for retrieval of the user profile.

If **accountName** is not specified or does not have a value and there is not an existing user profile for the **current user**, then a new user profile **MAY** [<11>](#) be created which represents the current user and the protocol server **MUST** send a response that includes zero **QuickLinkData** elements.

If **accountName** is not specified or does not have a value and there is an existing user profile for the current user, then the protocol server **MUST** return the **QuickLinkData** elements corresponding to the **quick link** properties of the current user.

If **accountName** is specified and does not match the login name of an existing user profile, the protocol server **MUST** send a **SOAP fault** message

#### 3.1.4.18.2.2 GetUserLinksResponse

The **GetUserLinksResponse** element is the body of the response **WSDL** message for the **GetUserLinks** WSDL operation.

```
<s:element name="GetUserLinksResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="GetUserLinksResult" type="tns:ArrayOfQuickLinkData"
        minOccurs="0" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**GetUserLinksResult:** A set of **QuickLinkData** (section [2.2.4.18](#)) elements corresponding to the **quick link** properties of the specified user.

If the **current user** is not the same user specified by **accountName**, and the current user is not authorized to administer the user profile service, then the protocol server **MUST** return a **SOAP fault** message.

The protocol server **MUST** only return **QuickLinkData** elements for quick links the current user has permission to view as specified in section [2.2.5.4](#).

If the user specified by **accountName** does not have any quick links, the response **MUST** contain zero **QuickLinkData** elements.

#### 3.1.4.19 GetUserMemberships

The **GetUserMemberships** **WSDL** operation is used to get the **memberships** associated with the specified **user profile**.

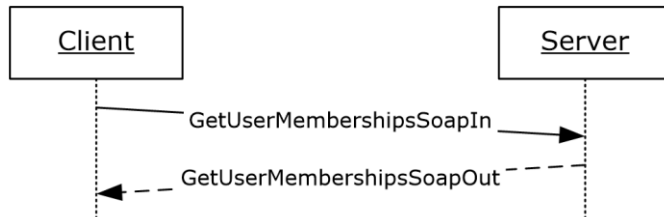
```

<wsdl:operation name="GetUserMemberships">
  <wsdl:input message="tns:GetUserMembershipsSoapIn"/>
  <wsdl:output message="tns:GetUserMembershipsSoapOut"/>
</wsdl:operation>

```

The protocol client sends a **GetUserMembershipsSoapIn** request WSDL message and the protocol server responds with a **GetUserMembershipsSoapOut** response WSDL message.

If the **current user** is not authorized to administer the user profile service and not requesting the profile of the current user, the protocol server **MUST** send a **SOAP fault** message.



**Figure 21: GetUserMemberships operation**

### 3.1.4.19.1 Messages

The following **WSDL** message definitions are specific to this operation.

#### 3.1.4.19.1.1 GetUserMembershipsSoapIn

The **GetUserMembershipsSoapIn** message is the request **WSDL** message for the **GetUserMemberships** WSDL operation.

The **SOAP action** value of the message is defined as:

```

http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetUserMemberships

```

The **SOAP body** contains a **GetUserMemberships** element.

#### 3.1.4.19.1.2 GetUserMembershipsSoapOut

The **GetUserMembershipsSoapOut** message is the response **WSDL** message for the **GetUserMemberships** WSDL operation.

The **SOAP body** contains a **GetUserMembershipsResponse** element.

### 3.1.4.19.2 Elements

The following **XML schema** element definitions are specific to this operation.

#### 3.1.4.19.2.1 GetUserMemberships

The **GetUserMemberships** element is the body of the request **WSDL** message for the **GetUserMemberships** WSDL operation.

```

<s:element name="GetUserMemberships">
  <s:complexType>
    <s:sequence>
      <s:element name="accountName" type="s:string" minOccurs="0" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>

```

```
</s:element>
```

**accountName:** A string that specifies the **login name** of the **user profile** to return the elements for. This value **MUST** be less than or equal to 400 characters in length. The protocol server **MUST** treat **accountName** in a case insensitive manner for retrieval of the user profile.

If **accountName** is not specified or does not have a value and there is not an existing user profile for the **current user**, then a new user profile **MAY** [<12>](#) be created which represents the current user and the protocol server **MUST** send a response that includes zero **MembershipData** elements.

If **accountName** is not specified or does not have a value and there is an existing user profile for the current user, then the protocol server **MUST** return elements for that user profile.

If **accountName** is specified and does not match the login name of an existing user profile, the protocol server **MUST** send a **SOAP fault** message.

### 3.1.4.19.2.2 GetUserMembershipsResponse

The **GetUserMembershipsResponse** element is the body of the response **WSDL** message for the **GetUserMemberships** WSDL operation.

```
<s:element name="GetUserMembershipsResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="GetUserMembershipsResult" type="tns:ArrayOfMembershipData"
        minOccurs="0" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**GetUserMembershipsResult:** A set of **MembershipData** (section [2.2.4.13](#)) elements corresponding to the specified user's **membership** properties.

If the **current user** is not the same user specified by **accountName**, and the current user is not authorized to administer the user profile service, then the protocol server **MUST** return a **SOAP fault** message.

The protocol server **MUST** only return **MembershipData** elements for memberships the current user has permission to view as specified in section [2.2.5.4](#).

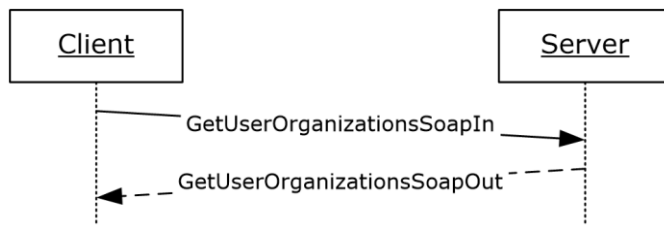
If the user specified by **accountName** does not have any memberships, the response **MUST** contain zero **MembershipData** elements.

### 3.1.4.20 GetUserOrganizations

The **GetUserOrganizations** **WSDL** operation is used to get the **organization** properties associated with the specified **user profile**.

```
<wsdl:operation name="GetUserOrganizations">
  <wsdl:input message="tns:GetUserOrganizationsSoapIn"/>
  <wsdl:output message="tns:GetUserOrganizationsSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **GetUserOrganizationsSoapIn** request WSDL message and the protocol server responds with a **GetUserOrganizationsSoapOut** response WSDL message.



**Figure 22: GetUserOrganizations operation**

### 3.1.4.20.1 Messages

The following **WSDL** message definitions are specific to this operation.

#### 3.1.4.20.1.1 GetUserOrganizationsSoapIn

The **GetUserOrganizationsSoapIn** message is the request **WSDL** message for the **GetUserOrganizations** WSDL operation.

The **SOAP action** value of the message is defined as:

```
http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetUserOrganizations
```

The **SOAP body** contains a **GetUserOrganizations** element.

#### 3.1.4.20.1.2 GetUserOrganizationsSoapOut

The **GetUserOrganizationsSoapOut** message is the response **WSDL** message for the **GetUserOrganizations** WSDL operation.

The **SOAP body** contains a **GetUserOrganizationsResponse** element.

### 3.1.4.20.2 Elements

The following **XML schema** element definitions are specific to this operation.

#### 3.1.4.20.2.1 GetUserOrganizations

The **GetUserOrganizations** element is the body of the request **WSDL** message for the **GetUserOrganizations** WSDL operation.

```

<s:element name="GetUserOrganizations">
  <s:complexType>
    <s:sequence>
      <s:element name="accountName" type="s:string" minOccurs="0" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
  
```

**accountName:** A string that specifies the **login name** of the **user profile** to return the elements for. This value **MUST** be less than or equal to 400 characters in length. The protocol server **MUST** treat **accountName** in a case insensitive manner for retrieval of the user profile. If **accountName** is not specified or does not have a value and there is not an existing user profile for the **current user**, the protocol server **MUST** send a **SOAP fault** message.

If **accountName** is not specified or does not have a value and there is an existing user profile for the current user, then the protocol server MUST return elements for that user profile.

If **accountName** is specified and does not match the login name of an existing user profile, the protocol server MUST send a SOAP fault message.

### 3.1.4.20.2.2 GetUserOrganizationsResponse

The **GetUserOrganizationsResponse** element is the body of the response **WSDL** message for the **GetUserOrganizations** WSDL operation.

```
<s:element name="GetUserOrganizationsResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="GetUserOrganizationsResult" type="tns:ArrayOfOrganizationProfileData"
        minOccurs="0" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**GetUserOrganizationsResult:** A set of **OrganizationProfileData** (section [2.2.4.14](#)) elements corresponding to the **organizations** of the specified user.

The protocol server MUST only return **OrganizationProfileData** elements for organizations the **current user** has permission to view. The protocol server determines the permission on each organization by checking the internal Privacy value for each as specified in section [2.2.5.4](#). However, these internally checked Privacy values are not returned as part of the response.

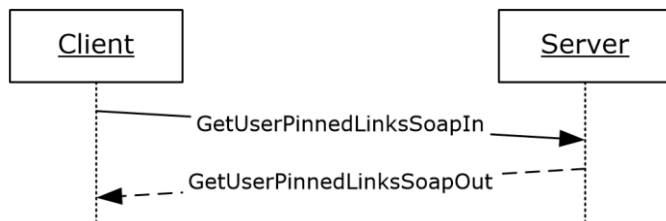
If the user specified by **accountName** does not have any organizations, the response MUST be an exception: "The object reference is not set to an instance of an object."

### 3.1.4.21 GetUserPinnedLinks

The **GetUserPinnedLinks** **WSDL** operation is used to get the **pinned links** associated with the specified **user profile**.

```
<wsdl:operation name="GetUserPinnedLinks">
  <wsdl:input message="tns:GetUserPinnedLinksSoapIn"/>
  <wsdl:output message="tns:GetUserPinnedLinksSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **GetUserPinnedLinksSoapIn** request WSDL message and the protocol server responds with a **GetUserPinnedLinksSoapOut** response WSDL message.



**Figure 23: GetUserPinnedLinks operation**

#### 3.1.4.21.1 Messages

The following **WSDL** message definitions are specific to this operation.

##### 3.1.4.21.1.1 GetUserPinnedLinksSoapIn

The **GetUserPinnedLinksSoapIn** message is the request **WSDL** message for the **GetUserPinnedLinks** WSDL operation.

The **SOAP action** value of the message is defined as:

```
http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetUserPinnedLinks
```

The **SOAP body** contains a **GetUserPinnedLinks** element.

### 3.1.4.21.1.2 GetUserPinnedLinksSoapOut

The **GetUserPinnedLinksSoapOut** message is the response **WSDL** message for the **GetUserPinnedLinks** WSDL operation.

The **SOAP body** contains a **GetUserPinnedLinksResponse** element.

### 3.1.4.21.2 Elements

The following **XML schema** element definitions are specific to this operation.

#### 3.1.4.21.2.1 GetUserPinnedLinks

The **GetUserPinnedLinks** element is the body of the request **WSDL** message for the **GetUserPinnedLinks** WSDL operation.

```
<s:element name="GetUserPinnedLinks">
  <s:complexType>
    <s:sequence>
      <s:element name="accountName" type="s:string" minOccurs="0" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**accountName:** A string that specifies the **login name** of the **user profile** to return the elements for. This value **MUST** be less than or equal to 400 characters in length. The protocol server **MUST** treat **accountName** in a case insensitive manner for retrieval of the user profile.

If **accountName** is not specified or does not have a value and there is not an existing user profile for the **current user**, then a new user profile **MUST** be created which represents the current user and the protocol server **MUST** send a response that includes zero **PinnedLinkData** elements.

If **accountName** is not specified or does not have a value and there is an existing user profile for the current user, then the protocol server **MUST** return elements for that user profile.

If **accountName** is specified and does not match the login name of an existing user profile, the protocol server **MUST** send a **SOAP fault** message.

#### 3.1.4.21.2.2 GetUserPinnedLinksResponse

The **GetUserPinnedLinksResponse** element is the body of the response **WSDL** message for the **GetUserPinnedLinks** WSDL operation.

```
<s:element name="GetUserPinnedLinksResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="GetUserPinnedLinksResult" type="tns:ArrayOfPinnedLinkData"
        minOccurs="0" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
```

```
</s:element>
```

**GetUserPinnedLinksResult:** A set of **PinnedLinkData** (section [2.2.4.15](#)) elements corresponding to the **pinned links** for the specified user.

The protocol server MUST only return **PinnedLinkData** elements for the pinned links the **current user** has permission to view as specified in section [2.2.5.4](#). For this function ONLY, the Privacy level is assumed to be Private.

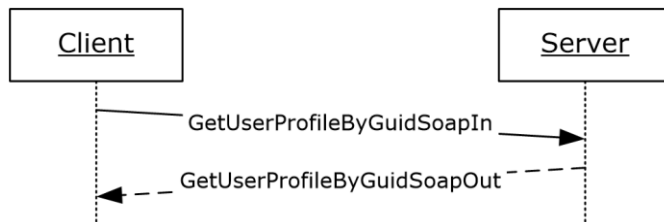
If the user specified by **accountName** does not have any pinned links, the response MUST contain zero **PinnedLinkData** elements.

### 3.1.4.22 GetUserProfileByGuid

The **GetUserProfileByGuid WSDL** operation is used to retrieve a **user profile** given the **GUID** of a user.

```
<wsdl:operation name="GetUserProfileByGuid">
  <wsdl:input message="tns:GetUserProfileByGuidSoapIn"/>
  <wsdl:output message="tns:GetUserProfileByGuidSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **GetUserProfileByGuidSoapIn** request WSDL message and the protocol server responds with a **GetUserProfileByGuidSoapOut** response WSDL message.



**Figure 24: GetUserProfileByGuid operation**

#### 3.1.4.22.1 Messages

The following **WSDL** message definitions are specific to this operation.

##### 3.1.4.22.1.1 GetUserProfileByGuidSoapIn

The **GetUserProfileByGuidSoapIn** message is the request **WSDL** message for the **GetUserProfileByGuid** WSDL operation.

The **SOAP action** value of the message is defined as:

```
http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetUserProfileByGuid
```

The **SOAP body** contains a **GetUserProfileByGuid** element.

##### 3.1.4.22.1.2 GetUserProfileByGuidSoapOut

The **GetUserProfileByGuidSoapOut** message is the response **WSDL** message from the **GetUserProfileByGuid** WSDL operation.

The **SOAP body** contains a **GetUserProfileByGuidResponse** element.

### 3.1.4.22.2 Elements

The following **XML schema** element definitions are specific to this operation.

#### 3.1.4.22.2.1 GetUserProfileByGuid

The **GetUserProfileByGuid** element is the body of the request **WSDL** message for the **GetUserProfileByGuid** WSDL operation.

```
<s:element name="GetUserProfileByGuid">
  <s:complexType>
    <s:sequence>
      <s:element name="guid" type="s1:guid"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**guid:** A **guid** type that specifies the **GUID** associated with the user for which to retrieve the **user profile**.

If the GUID does not exist indicating that it does not refer to an existing user, the protocol server MUST send a **SOAP fault** message.

#### 3.1.4.22.2.2 GetUserProfileByGuidResponse

The **GetUserProfileByGuidResponse** element is the body of the response **WSDL** message for the **GetUserProfileByGuid** WSDL operation.

```
<s:element name="GetUserProfileByGuidResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="GetUserProfileByGuidResult"
type="tns:ArrayOfPropertyData"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**GetUserProfileByGuidResult:** A set of **PropertyData** (section [2.2.4.16](#)) elements corresponding to the properties of the user specified by **guid**.

If the **current user** is not the same user specified by **guid**, and the current user is not authorized to administer the user profile service, then the protocol server MUST set the **Privacy** element of any **PropertyData** elements returned to "NotSet".

The protocol server MUST only return **PropertyData** elements for properties the current user has permission to view as specified in section [2.2.5.4](#).

If the user specified by **guid** does not have any properties, the response MUST contain zero **PropertyData** elements.

### 3.1.4.23 GetUserProfileByIndex

The **GetUserProfileByIndex** **WSDL** operation is used to retrieve a **user profile** given a unique identifier. This operation can be used to iterate through all of the existing user profiles.

```
<wsdl:operation name="GetUserProfileByIndex">
```

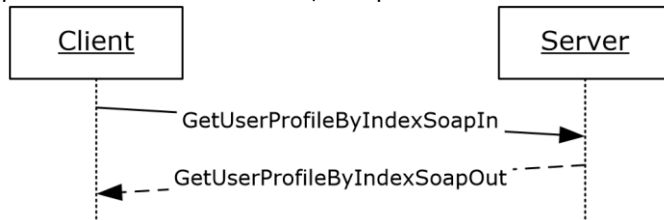


```

    <wsdl:input message="tns:GetUserProfileByIndexSoapIn"/>
    <wsdl:output message="tns:GetUserProfileByIndexSoapOut"/>
  </wsdl:operation>

```

The protocol client sends a **GetUserProfileByIndexSoapIn** request WSDL message and the protocol server responds with a **GetUserProfileByIndexSoapOut** response WSDL message. If the **current user** is not authorized to administer the user profile service and not requesting the profile of the current user, the protocol server **MUST** send a **SOAP fault** message.



**Figure 25: GetUserProfileByIndex operation**

### 3.1.4.23.1 Messages

The following **WSDL** message definitions are specific to this operation.

#### 3.1.4.23.1.1 GetUserProfileByIndexSoapIn

The **GetUserProfileByIndexSoapIn** message is the request **WSDL** message for the **GetUserProfileByIndex** WSDL operation.

The **SOAP action** value of the message is defined as:

```

http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetUserProfileByIndex

```

The **SOAP body** contains a **GetUserProfileByIndex** element.

#### 3.1.4.23.1.2 GetUserProfileByIndexSoapOut

The **GetUserProfileByIndexSoapOut** message is the response **WSDL** message for the **GetUserProfileByIndex** WSDL operation.

The **SOAP body** contains a **GetUserProfileByIndexResponse** element.

### 3.1.4.23.2 Elements

The following **XML schema** element definitions are specific to this operation.

#### 3.1.4.23.2.1 GetUserProfileByIndex

The **GetUserProfileByIndex** element is the body of the request **WSDL** message for the **GetUserProfileByIndex** WSDL operation.

```

<s:element name="GetUserProfileByIndex">
  <s:complexType>
    <s:sequence>
      <s:element name="index" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>

```

**index:** An **int** type that specifies which **user profile** is returned. The protocol server MUST return the user profile whose index matches the smallest user profile index that is greater than index.

If index is greater than the indices of all existing user profiles, the protocol server MUST return a `GetUserProfileByIndexResult` element, in which `NextValue` is set to -1 and all other fields are null.

### 3.1.4.23.2 GetUserProfileByIndexResponse

The **GetUserProfileByIndexResponse** element is the body of the response **WSDL** message for the **GetUserProfileByIndex** WSDL operation.

```
<s:element name="GetUserProfileByIndexResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="GetUserProfileByIndexResult"
        type="tns:GetUserProfileByIndexResult" minOccurs="0" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**GetUserProfileByIndexResult:** A complex type that represents the data of a user profile (section [3.1.4.23.3.1](#)) as retrieved by the `GetUserProfileByIndex` WSDL operation (section [3.1.4.23.2.1](#))

The protocol server MUST return a `GetUserProfileByIndexResult` (section 3.1.4.23.3.1) in the response WSDL message as long as it is not returning a **SOAP fault**.

### 3.1.4.23.3 Complex Types

The following **XML schema** complex type definitions are specific to this operation.

#### 3.1.4.23.3.1 GetUserProfileByIndexResult

The **GetUserProfileByIndexResult** complex type is used to represent the returned **user profile** data.

```
<s:complexType name="GetUserProfileByIndexResult">
  <s:sequence>
    <s:element name="NextValue" type="s:string" minOccurs="0" maxOccurs="1"/>
    <s:element name="UserProfile" type="tns:ArrayOfPropertyData" minOccurs="0"
      maxOccurs="1"/>
    <s:element name="Colleagues" type="tns:ArrayOfContactData" minOccurs="0"
      maxOccurs="1"/>
    <s:element name="QuickLinks" type="tns:ArrayOfQuickLinkData" minOccurs="0"
      maxOccurs="1"/>
    <s:element name="PinnedLinks" type="tns:ArrayOfPinnedLinkData" minOccurs="0"
      maxOccurs="1"/>
    <s:element name="Memberships" type="tns:ArrayOfMembershipData" minOccurs="0"
      maxOccurs="1"/>
  </s:sequence>
</s:complexType>
```

**NextValue:** A string that specifies the user profile index of the next user profile that is greater than index. This value MUST uniquely identify another user profile object, or be set to -1 if no such object exists. The value inside the string is an integer of the same type as that accepted by `GetUserProfileByIndexResult`.

**UserProfile:** A set of **PropertyData** (section [2.2.4.16](#)) elements corresponding to the properties of the specified user.

**Colleagues:** A set of **ContactData** (section [2.2.4.11](#)) elements corresponding to the colleague properties of the specified user.

**QuickLinks:** A set of **QuickLinkData** (section [2.2.4.18](#)) elements corresponding to the **quick link** properties of the specified user.

**PinnedLinks:** A set of **PinnedLinkData** (section [2.2.4.15](#)) elements corresponding to the **pinned links** of the specified user.

**Memberships:** A set of **MembershipData** (section [2.2.4.13](#)) elements corresponding to the **memberships** of the specified user.

If the **current user** is not the same as the user retrieved through `GetUserProfileByIndex` using **index**, and the current user is not authorized to administer the user profile service, the server must return a **SOAP fault** message.

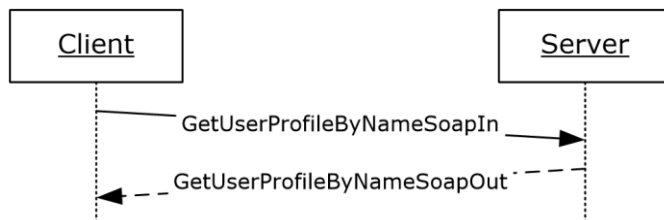
The protocol server **MUST** only return elements for properties the current user has permission to view as specified in section [2.2.5.4](#).

### 3.1.4.24 GetUserProfileByName

The **GetUserProfileByName** **WSDL** operation is used to retrieve a **user profile** given a **login name**.

```
<wsdl:operation name="GetUserProfileByName">
  <wsdl:input message="tns:GetUserProfileByNameSoapIn"/>
  <wsdl:output message="tns:GetUserProfileByNameSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **GetUserProfileByNameSoapIn** request WSDL message and the protocol server responds with a **GetUserProfileByNameSoapOut** response WSDL message.



**Figure 26: GetUserProfileByName operation**

#### 3.1.4.24.1 Messages

The following **WSDL** message definitions are specific to this operation.

##### 3.1.4.24.1.1 GetUserProfileByNameSoapIn

The **GetUserProfileByNameSoapIn** message is the request **WSDL** message for the **GetUserProfileByName** WSDL operation.

The **SOAP action** value of the message is defined as:

```
http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetUserProfileByName
```

The **SOAP body** contains a **GetUserProfileByName** element.

### 3.1.4.24.1.2 GetUserProfileByNameSoapOut

The **GetUserProfileByNameSoapOut** message is the response **WSDL** message for the **GetUserProfileByName** WSDL operation.

The **SOAP body** contains a **GetUserProfileByNameResponse** element.

### 3.1.4.24.2 Elements

The following **XML schema** element definitions are specific to this operation.

#### 3.1.4.24.2.1 GetUserProfileByName

The **GetUserProfileByName** element is the body of the request **WSDL** message for the **GetUserProfileByName** WSDL operation.

```
<s:element name="GetUserProfileByName">
  <s:complexType>
    <s:sequence>
      <s:element name="accountName" minOccurs="0" maxOccurs="1" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**accountName:** A string that specifies the **login name** of the **user profile** to return the elements for. This value **MUST** be less than or equal to 400 characters in length. The protocol server **MUST** treat **accountName** in a case insensitive manner for retrieval of the user profile.

If **accountName** is an empty string or NULL, and there is not an existing user profile for the **current user**, then a new user profile **MUST** be created which represents the user and the protocol server **MUST** return that user profile.

If **accountName** is an empty string or NULL, and there is an existing user profile for the current user, then the user profile for the current user **MUST** be returned from the server.

If **accountName** is specified and does not match the login name of an existing user profile, the protocol server **MUST** send a **SOAP fault** message.

#### 3.1.4.24.2.2 GetUserProfileByNameResponse

The **GetUserProfileByNameResponse** element is the body of the response **WSDL** message for the **GetUserProfileByName** WSDL operation.

```
<s:element name="GetUserProfileByNameResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="GetUserProfileByNameResult"
type="tns:ArrayOfPropertyData"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**GetUserProfileByNameResult:** A set of **PropertyData** (section [2.2.4.16](#)) elements corresponding to the properties of the user specified by **accountName**.

If the **current user** is not the same user specified by **accountName**, and the current user is not authorized to administer the user profile service, then the protocol server **MUST** set the **Privacy** element of any **PropertyData** elements returned to "NotSet".

The protocol server MUST only return **PropertyData** elements for properties the current user has permission to view as specified in section [2.2.5.4](#).

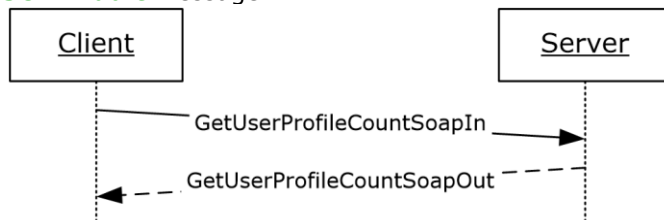
If the user specified by **accountName** does not have any properties, the response MUST contain zero **PropertyData** elements.

### 3.1.4.25 GetUserProfileCount

The **GetUserProfileCount** **WSDL** operation is used return the number of **user profiles** stored by the protocol server.

```
<wsdl:operation name="GetUserProfileCount">
  <wsdl:input message="tns:GetUserProfileCountSoapIn"/>
  <wsdl:output message="tns:GetUserProfileCountSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **GetUserProfileCountSoapIn** request WSDL message and the protocol server responds with a **GetUserProfileCountSoapOut** response WSDL message. If the user is not authorized to administer the user profile service, the protocol server MUST send a **SOAP fault** message.



**Figure 27: GetUserProfileCount operation**

#### 3.1.4.25.1 Messages

The following **WSDL** message definitions are specific to this operation.

##### 3.1.4.25.1.1 GetUserProfileCountSoapIn

The **GetUserProfileCountSoapIn** message is the request **WSDL** message for the **GetUserProfileCount** WSDL operation.

The **SOAP action** value of the message is defined as:

```
http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetUserProfileCount
```

The **SOAP body** contains a **GetUserProfileCount** element.

##### 3.1.4.25.1.2 GetUserProfileCountSoapOut

The **GetUserProfileCountSoapOut** message is the response **WSDL** message for the **GetUserProfileCount** WSDL operation.

The **SOAP body** contains a **GetUserProfileCountResponse** element.

#### 3.1.4.25.2 Elements

The following **XML schema** element definitions are specific to this operation.

##### 3.1.4.25.2.1 GetUserProfileCount

The **GetUserProfileCount** element is the body of the request **WSDL** message for the **GetUserProfileCount** WSDL operation.

```
<s:element name="GetUserProfileCount">
  <s:complexType/>
</s:element>
```

### 3.1.4.25.2.2 GetUserProfileCountResponse

The **GetUserProfileCountResponse** element is the body of the response **WSDL** message for the **GetUserProfileCount** WSDL operation.

```
<s:element name="GetUserProfileCountResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="GetUserProfileCountResult" type="s:long"
    />
  </s:sequence>
</s:complexType>
</s:element>
```

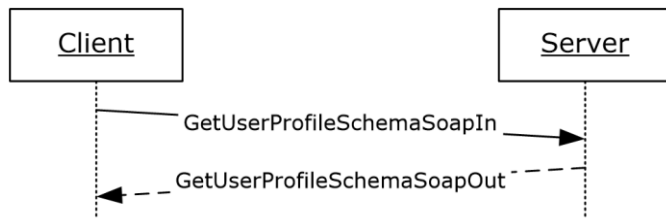
**GetUserProfileCountResult:** A **long** type that specifies the number of **user profiles** on the host server.

### 3.1.4.26 GetUserProfileSchema

The **GetUserProfileSchema** **WSDL** operation is used to retrieve schema information about each of the defined properties.

```
<wsdl:operation name="GetUserProfileSchema">
  <wsdl:input message="tns:GetUserProfileSchemaSoapIn"/>
  <wsdl:output message="tns:GetUserProfileSchemaSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **GetUserProfileSchemaSoapIn** request WSDL message and the protocol server responds with a **GetUserProfileSchemaSoapOut** response WSDL message.



**Figure 28: GetUserProfileSchema operation**

#### 3.1.4.26.1 Messages

The following **WSDL** message definitions are specific to this operation.

##### 3.1.4.26.1.1 GetUserProfileSchemaSoapIn

The **GetUserProfileSchemaSoapIn** message is the request **WSDL** message for the **GetUserProfileSchema** WSDL operation.

The **SOAP action** value of the message is defined as:

```
http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetUserProfileSchema
```

The **SOAP body** contains a **GetUserProfileSchema** element.

#### 3.1.4.26.1.2 GetUserProfileSchemaSoapOut

The **GetUserProfileSchemaSoapOut** message is the response **WSDL** message for the **GetUserProfileSchema** WSDL operation.

The **SOAP body** contains a **GetUserProfileSchemaResponse** element.

#### 3.1.4.26.2 Elements

The following **XML schema** element definitions are specific to this operation.

##### 3.1.4.26.2.1 GetUserProfileSchema

The **GetUserProfileSchema** element is the body of the request **WSDL** message for the **GetUserProfileSchema** WSDL operation.

```
<s:element name="GetUserProfileSchema">
  <s:complexType/>
</s:element>
```

##### 3.1.4.26.2.2 GetUserProfileSchemaResponse

The **GetUserProfileSchemaResponse** element is the body of the response **WSDL** message for the **GetUserProfileSchema** WSDL operation.

```
<s:element name="GetUserProfileSchemaResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="GetUserProfileSchemaResult"
type="tns:ArrayOfPropertyInfo" />
    </s:sequence>
  </s:complexType>
</s:element>
```

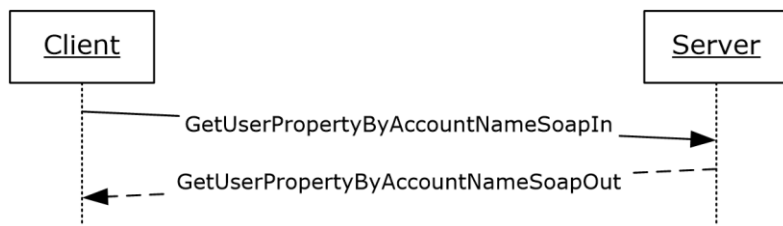
**GetUserProfileSchemaResult:** A set of **PropertyInfo** (section [2.2.4.17](#)) elements that is a listing of the properties of a user profile and of the attributes of each property.

#### 3.1.4.27 GetUserPropertyByAccountName

The **GetUserPropertyByAccountName** **WSDL** operation is used to get the value of a specific property for the specified user.

```
<wsdl:operation name="GetUserPropertyByAccountName">
  <wsdl:input message="tns:GetUserPropertyByAccountNameSoapIn"/>
  <wsdl:output message="tns:GetUserPropertyByAccountNameSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **GetUserPropertyByAccountNameSoapIn** request WSDL message and the protocol server responds with a **GetUserPropertyByAccountNameSoapIn** response WSDL message.



**Figure 29: GetUserPropertyByAccountName operation**

### 3.1.4.27.1 Messages

The following **WSDL** message definitions are specific to this operation.

#### 3.1.4.27.1.1 GetUserPropertyByAccountNameSoapIn

The **GetUserPropertyByAccountNameSoapIn** message is the request **WSDL** message for the **GetUserPropertyByAccountName** WSDL operation.

The **SOAP action** value of the message is defined as:

```
http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetUserPropertyByAccountName
```

The **SOAP body** contains a **GetUserPropertyByAccountName** element.

#### 3.1.4.27.1.2 GetUserPropertyByAccountNameSoapOut

The **GetUserPropertyByAccountNameSoapOut** message is the response **WSDL** message for the **GetUserPropertyByAccountName** WSDL operation.

The **SOAP body** contains a **GetUserPropertyByAccountNameResponse** element.

### 3.1.4.27.2 Elements

The following **XML schema** element definitions are specific to this operation.

#### 3.1.4.27.2.1 GetUserPropertyByAccountName

The **GetUserPropertyByAccountName** element is the body of the request **WSDL** message for the **GetUserPropertyByAccountName** WSDL operation.

```

<s:element name="GetUserPropertyByAccountName">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="accountName" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
  
```

**accountName:** A string that specifies the **login name** of the **user profile** for which to get the property. This value **MUST** be less than or equal to 400 characters in length. The protocol server **MUST** treat **accountName** in a case insensitive manner for retrieval of the user profile.



If **accountName** is not specified or does not have a value and there is not an existing user profile for the **current user**, then a new user profile MUST be created which represents the user and the protocol server MUST use that user profile to get the property.

If **accountName** is not specified or does not have a value and there is an existing user profile for the current user, then the protocol server MUST use that user profile to get the property.

If **accountName** is specified and does not match the login name of an existing user profile, the protocol server MUST send a **SOAP fault** message.

**propertyName**: A string that specifies the name of an element that belongs to the **PropertyData** (section [2.2.4.16](#)) complex type.

If **PropertyData** does not have a property name specified by **propertyName**, the protocol server MUST send a SOAP fault message.

### 3.1.4.27.2.2 GetUserPropertyByAccountNameResponse

The **GetUserPropertyByAccountNameResponse** element is the body of the response **WSDL** message for the **GetUserPropertyByAccountName** WSDL operation.

```
<s:element name="GetUserPropertyByAccountNameResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="GetUserPropertyByAccountNameResult"
type="tns:PropertyData" />
    </s:sequence>
  </s:complexType>
</s:element>
```

**GetUserPropertyByAccountNameResult**: A **PropertyData** element containing the specified user profile property of the user with the specified **login name**.

If the **current user** is not the same user specified by **accountName**, and the current user is not authorized to administer the user profile service, then the protocol server MUST send a **SOAP fault** message.

The protocol server MUST only return a **PropertyData** element for a property that the current user has permission to view as specified in section [2.2.5.4](#).

If **propertyName** is not specified or does not have a value, the response MUST contain zero **PropertyData** elements.

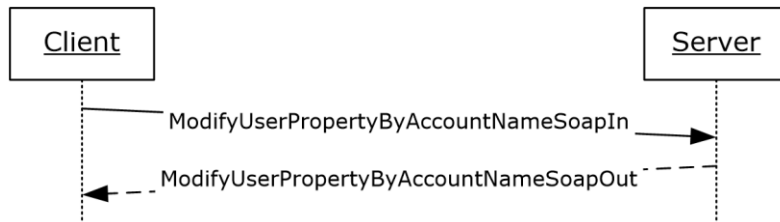
### 3.1.4.28 ModifyUserPropertyByAccountName

The **ModifyUserPropertyByAccountName** **WSDL** operation is used to modify a set of properties for the specified user.

```
<wsdl:operation name="ModifyUserPropertyByAccountName">
  <wsdl:input message="tns:ModifyUserPropertyByAccountNameSoapIn"/>
  <wsdl:output message="tns:ModifyUserPropertyByAccountNameSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **ModifyUserPropertyByAccountNameSoapIn** request WSDL message and the protocol server responds with a **ModifyUserPropertyByAccountNameSoapOut** response WSDL message.

If the user is not authorized to administer the user profile service and is not updating his or her own **user profile**, the protocol server MUST send a **SOAP fault** message.



**Figure 30: ModifyUserPropertyByAccountName operation**

### 3.1.4.28.1 Messages

The following **WSDL** message definitions are specific to this operation.

#### 3.1.4.28.1.1 ModifyUserPropertyByAccountNameSoapIn

The **ModifyUserPropertyByAccountNameSoapIn** message is the request **WSDL** message for the **ModifyUserPropertyByAccountName** WSDL operation.

The **SOAP action** value of the message is defined as:

```
http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/ModifyUserPropertyByAccountName
```

The **SOAP body** contains a **ModifyUserPropertyByAccountName** element.

#### 3.1.4.28.1.2 ModifyUserPropertyByAccountNameSoapOut

The **ModifyUserPropertyByAccountNameSoapOut** message is the response **WSDL** message for the **ModifyUserPropertyByAccountName** WSDL operation.

The **SOAP body** contains a **ModifyUserPropertyByAccountNameResponse** element.

### 3.1.4.28.2 Elements

The following **XML schema** element definitions are specific to this operation.

#### 3.1.4.28.2.1 ModifyUserPropertyByAccountName

The **ModifyUserPropertyByAccountName** element updates a set of **PropertyData** (section [2.2.4.16](#)) elements for the specified user.

```
<s:element name="ModifyUserPropertyByAccountName">
  <s:complexType>
    <s:sequence>
      <s:element name="accountName" type="s:string" minOccurs="0" maxOccurs="1"/>
      <s:element name="newData" type="tns:ArrayOfPropertyData" minOccurs="0"
        maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**accountName:** A string that specifies the **login name** of the **user profile** to return the elements for. This value MUST be less than or equal to 400 characters in length. The protocol server MUST treat **accountName** in a case insensitive manner for retrieval of the user profile.

If **accountName** is not specified or does not have a value and there is not an existing user profile for the **current user**, then a new user profile MAY<13> be created which represents the current user. If **accountName** is not specified or does not have a value and there is an existing user profile for the current user, then the protocol server MUST return elements for that user profile. If **accountName** is specified and does not match the login name of an existing user profile, the protocol server MUST send a **SOAP fault** message.

**newData:** A set of **PropertyData** (section 2.2.4.16) elements that contain the properties to modify.

If a **PropertyData** element with the name "UserProfile\_GUID" is passed in with a **GUID** that matches a user, and **accountName** is NULL or an empty string, then that user MUST be modified. If the GUID does not match an existing user, the protocol server MUST send a SOAP fault message. Additionally, for the properties specified by **newData**, if the property is not editable, then the property MUST NOT be updated. For properties that are only editable by users authorized to administer the user profile service, the current user MUST be authorized to administer the user profile service to update the property.

### 3.1.4.28.2 ModifyUserPropertyByAccountNameResponse

The **ModifyUserPropertyByAccountNameResponse** element is the body of the response **WSDL** message for the **ModifyUserPropertyByAccountName** WSDL operation.

```
<s:element name="ModifyUserPropertyByAccountNameResponse">
  <s:complexType/>
</s:element>
```

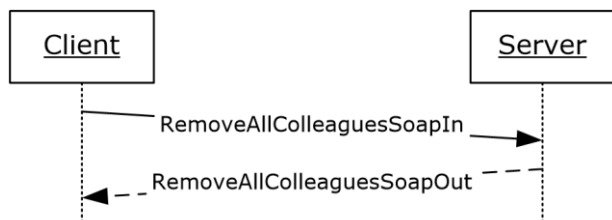
### 3.1.4.29 RemoveAllColleagues

The **RemoveAllColleagues** **WSDL** operation is used to update the specified **user profile** by deleting all colleague associations present in the user profile.

```
<wsdl:operation name="RemoveAllColleagues">
  <wsdl:input message="tns:RemoveAllColleaguesSoapIn"/>
  <wsdl:output message="tns:RemoveAllColleaguesSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **RemoveAllColleaguesSoapIn** request WSDL message and the protocol server responds with a **RemoveAllColleaguesSoapOut** response WSDL message.

If the user is not authorized to administer the user profile service and is not updating his or her own user profile, the protocol server MUST send a **SOAP fault** message.



**Figure 31: RemoveAllColleagues operation**

#### 3.1.4.29.1 Messages

The following **WSDL** message definitions are specific to this operation.

##### 3.1.4.29.1.1 RemoveAllColleaguesSoapIn

The **RemoveAllColleaguesSoapIn** message is the request **WSDL** message for the **RemoveAllColleagues** WSDL operation.

The **SOAP action** value of the message is defined as:

```
http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/RemoveAllColleagues
```

The **SOAP body** contains a **RemoveAllColleagues** element.

#### 3.1.4.29.1.2 RemoveAllColleaguesSoapOut

The **RemoveAllColleaguesSoapOut** message is the response **WSDL** message for the **RemoveAllColleagues** WSDL operation.

The **SOAP body** contains a **RemoveAllColleaguesResponse** element.

#### 3.1.4.29.2 Elements

The following **XML schema** element definitions are specific to this operation.

##### 3.1.4.29.2.1 RemoveAllColleagues

The **RemoveAllColleagues** element is the body of the request **WSDL** message for the **RemoveAllColleagues** WSDL operation.

```
<s:element name="RemoveAllColleagues">
  <s:complexType>
    <s:sequence>
      <s:element name="accountName" type="s:string" minOccurs="0" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**accountName:** A string that specifies the **login name** of the **user profile** to be updated. This value MUST be less than or equal to 400 characters in length. If this element is not specified or does not have a value, the protocol server MUST use the user profile of the **current user**, and return a **SOAP fault** message if the current user does not have a user profile. The protocol server MUST treat **accountName** in a case insensitive manner for retrieval of the user profile. If **accountName** does not correspond to an existing user profile, the protocol server MUST send a SOAP fault message.

##### 3.1.4.29.2.2 RemoveAllColleaguesResponse

The **RemoveAllColleaguesResponse** element is the body of the response **WSDL** message for the **RemoveAllColleagues** WSDL operation.

```
<s:element name="RemoveAllColleaguesResponse">
  <s:complexType/>
</s:element>
```

#### 3.1.4.30 RemoveAllLinks

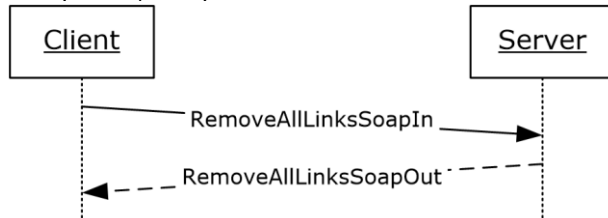
The **RemoveAllLinks** **WSDL** operation is used to update the specified **user profile** by deleting all **quick link** properties present in the user profile of the specified user.

```

<wsdl:operation name="RemoveAllLinks">
  <wsdl:input message="tns:RemoveAllLinksSoapIn"/>
  <wsdl:output message="tns:RemoveAllLinksSoapOut"/>
</wsdl:operation>

```

The protocol client sends a **RemoveAllLinksSoapIn** request WSDL message and the protocol server responds with a **RemoveAllLinksSoapOut** response WSDL message. If the user is not authorized to administer the user profile service and is not updating his or her own user profile, the protocol server MUST send a **SOAP fault** message.



**Figure 32: RemoveAllLinks operation**

### 3.1.4.30.1 Messages

The following **WSDL** message definitions are specific to this operation.

#### 3.1.4.30.1.1 RemoveAllLinksSoapIn

The **RemoveAllLinksSoapIn** message is the request **WSDL** message for the **RemoveAllLinks** WSDL operation.

The **SOAP action** value of the message is defined as:

```
http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/RemoveAllLinks
```

The **SOAP body** contains a **RemoveAllLinks** element.

#### 3.1.4.30.1.2 RemoveAllLinksSoapOut

The **RemoveAllLinksSoapOut** message is the response **WSDL** message for the **RemoveAllLinks** WSDL operation.

The **SOAP body** contains a **RemoveAllLinksResponse** element.

### 3.1.4.30.2 Elements

The following **XML schema** element definitions are specific to this operation.

#### 3.1.4.30.2.1 RemoveAllLinks

The **RemoveAllLinks** element is the body of the request **WSDL** message for the **RemoveAllLinks** WSDL operation.

```

<s:element name="RemoveAllLinks">
  <s:complexType>
    <s:sequence>
      <s:element name="accountName" type="s:string" minOccurs="0" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>

```

```
</s:element>
```

**accountName:** A string that specifies the **login name** of the **user profile** to be updated. This value MUST be less than or equal to 400 characters in length. If this element is not specified or does not have a value, the protocol server MUST use the user profile of the **current user**, and return a **SOAP fault** message if the current user does not have a user profile. The protocol server MUST treat **accountName** in a case insensitive manner for retrieval of the user profile.

If **accountName** does not correspond to an existing user profile, the protocol server MUST send a SOAP fault message.

### 3.1.4.30.2.2 RemoveAllLinksResponse

The **RemoveAllLinksResponse** element is the body of the response **WSDL** message for the **RemoveAllLinks** WSDL operation.

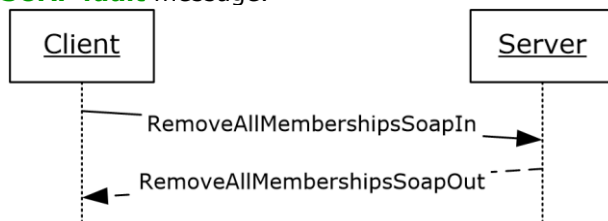
```
<s:element name="RemoveAllLinksResponse">
  <s:complexType/>
</s:element>
```

### 3.1.4.31 RemoveAllMemberships

The **RemoveAllMemberships** **WSDL** operation is used to update the specified **user profile** by deleting all **memberships** present in the user profile.

```
<wsdl:operation name="RemoveAllMemberships">
  <wsdl:input message="tns:RemoveAllMembershipsSoapIn"/>
  <wsdl:output message="tns:RemoveAllMembershipsSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **RemoveAllMembershipsSoapIn** request WSDL message and the protocol server responds with a **RemoveAllMembershipsSoapOut** response WSDL message. If the user is not authorized to administer the user profile service, the protocol server MUST send a **SOAP fault** message.



**Figure 33: RemoveAllMemberships operation**

#### 3.1.4.31.1 Messages

The following **WSDL** message definitions are specific to this operation.

##### 3.1.4.31.1.1 RemoveAllMembershipsSoapIn

The **RemoveAllMembershipsSoapIn** message is the request **WSDL** message for the **RemoveAllMemberships** WSDL operation.

The **SOAP action** value of the message is defined as:

http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/RemoveAllMemberships

The **SOAP body** contains a **RemoveAllMemberships** element.

#### 3.1.4.31.1.2 RemoveAllMembershipsSoapOut

The **RemoveAllMembershipsSoapOut** message is the response **WSDL** message for the **RemoveAllMemberships** WSDL operation.

The **SOAP body** contains a **RemoveAllMembershipsResponse** element.

#### 3.1.4.31.2 Elements

The following **XML schema** element definitions are specific to this operation.

##### 3.1.4.31.2.1 RemoveAllMemberships

The **RemoveAllMemberships** element is the body of the request **WSDL** message for the **RemoveAllMemberships** WSDL operation.

```
<s:element name="RemoveAllMemberships">
  <s:complexType>
    <s:sequence>
      <s:element name="accountName" type="s:string" minOccurs="0" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**accountName:** A string that specifies the **login name** of the **user profile** to be updated. This value MUST be less than or equal to 400 characters in length. If this element is not specified or does not have a value, the protocol server MUST use the user profile of the **current user**, and return a **SOAP fault** message if the current user does not have a user profile. The protocol server MUST treat **accountName** in a case insensitive manner for retrieval of the user profile.

If **accountName** does not correspond to an existing user profile, the protocol server MUST send a SOAP fault message. For memberships, only administrators or profile owner can add or delete items.

##### 3.1.4.31.2.2 RemoveAllMembershipsResponse

The **RemoveAllMembershipsResponse** element is the body of the response **WSDL** message for the **RemoveAllMemberships** WSDL operation.

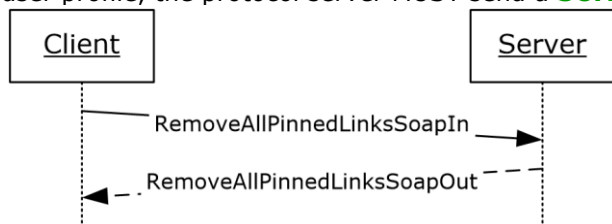
```
<s:element name="RemoveAllMembershipsResponse">
  <s:complexType/>
</s:element>
```

#### 3.1.4.32 RemoveAllPinnedLinks

The **RemoveAllPinnedLinks** **WSDL** operation is used to update the specified **user profile** by deleting all **pinned link** properties present in the user profile.

```
<wsdl:operation name="RemoveAllPinnedLinks">
  <wsdl:input message="tns:RemoveAllPinnedLinksSoapIn"/>
  <wsdl:output message="tns:RemoveAllPinnedLinksSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **RemoveAllPinnedLinksSoapIn** request WSDL message and the protocol server responds with a **RemoveAllPinnedLinksSoapOut** response WSDL message. If the user is not authorized to administer the user profile service and is not updating his or her own user profile, the protocol server MUST send a **SOAP fault** message.



**Figure 34: RemoveAllPinnedLinks operation**

### 3.1.4.32.1 Messages

The following **WSDL** message definitions are specific to this operation.

#### 3.1.4.32.1.1 RemoveAllPinnedLinksSoapIn

The **RemoveAllPinnedLinksSoapIn** message is the request **WSDL** message for the **RemoveAllPinnedLinks** WSDL operation.

The **SOAP action** value of the message is defined as:

```
http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/RemoveAllPinnedLinks
```

The **SOAP body** contains a **RemoveAllPinnedLinks** element.

#### 3.1.4.32.1.2 RemoveAllPinnedLinksSoapOut

The **RemoveAllPinnedLinksSoapOut** message is the response **WSDL** message for the **RemoveAllPinnedLinks** WSDL operation.

The **SOAP body** contains a **RemoveAllPinnedLinksResponse** element.

### 3.1.4.32.2 Elements

The following **XML schema** element definitions are specific to this operation.

#### 3.1.4.32.2.1 RemoveAllPinnedLinks

The **RemoveAllPinnedLinks** element is the body of the request **WSDL** message for the **RemoveAllPinnedLinks** WSDL operation.

```

<s:element name="RemoveAllPinnedLinks">
  <s:complexType>
    <s:sequence>
      <s:element name="accountName" type="s:string" minOccurs="0" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
  
```

**accountName:** A string that specifies the **login name** of the **user profile** to be updated. This value MUST be less than or equal to 400 characters in length. The protocol server MUST treat **accountName** in a case insensitive manner for retrieval of the user profile.



If **accountName** is not specified or does not have a value and there is not an existing user profile for the **current user**, then a new user profile MAY<14> be created which represents the current user and the protocol server MUST update that user profile.

If **accountName** is not specified or does not have a value and there is an existing user profile for the current user, then the protocol server MUST update that user profile.

If **accountName** is specified and does not match the login name of an existing user profile, then the protocol server MUST send a **SOAP fault** message.

### 3.1.4.32.2 RemoveAllPinnedLinksResponse

The **RemoveAllPinnedLinksResponse** element is the body of the response **WSDL** message for the **RemoveAllPinnedLinks** WSDL operation.

```
<s:element name="RemoveAllPinnedLinksResponse">
  <s:complexType/>
</s:element>
```

### 3.1.4.33 RemoveColleague

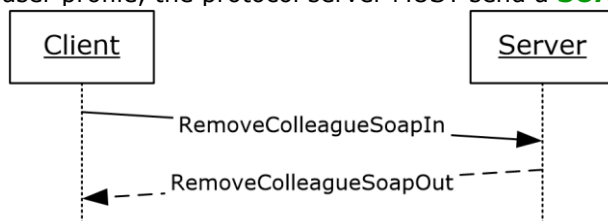
The **RemoveColleague** **WSDL** operation is used to update the specified **user profile** by deleting a specified colleague property association currently present in the user profile.

The operation of removing a colleague is unidirectional, meaning that when a colleague of the specified user is removed, the user profile information of the colleague being removed is not changed.

```
<wsdl:operation name="RemoveColleague">
  <wsdl:input message="tns:RemoveColleagueSoapIn"/>
  <wsdl:output message="tns:RemoveColleagueSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **RemoveColleagueSoapIn** request WSDL message and the protocol server responds with a **RemoveColleagueSoapOut** response WSDL message.

If the user is not authorized to administer the user profile service and is not updating his or her own user profile, the protocol server MUST send a **SOAP fault** message.



**Figure 35: RemoveColleague operation**

#### 3.1.4.33.1 Messages

The following **WSDL** message definitions are specific to this operation.

##### 3.1.4.33.1.1 RemoveColleagueSoapIn

The **RemoveColleagueSoapIn** message is the request **WSDL** message for the **RemoveColleague** WSDL operation.

The **SOAP action** value of the message is defined as:

```
http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/RemoveColleague
```

The **SOAP body** contains a **RemoveColleague** element.

### 3.1.4.33.1.2 RemoveColleagueSoapOut

The **RemoveColleagueSoapOut** message is the response **WSDL** message for the **RemoveColleague** WSDL operation.

The **SOAP body** contains a RemoveColleagueResponse element.

### 3.1.4.33.2 Elements

The following **XML schema** element definitions are specific to this operation.

#### 3.1.4.33.2.1 RemoveColleague

The **RemoveColleague** element is the body of the request **WSDL** message for the **RemoveColleague** WSDL operation.

```
<s:element name="RemoveColleague">
  <s:complexType>
    <s:sequence>
      <s:element name="accountName" type="s:string" minOccurs="0" maxOccurs="1"/>
      <s:element name="colleagueAccountName" type="s:string" minOccurs="0"
        maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**accountName:** A string that specifies the **login name** of the **user profile** to be updated. This value MUST be less than or equal to 400 characters in length. If this element is not specified or does not have a value, the protocol server MUST use the user profile of the **current user**, and return a **SOAP fault** message if the current user does not have a user profile. The protocol server MUST treat **accountName** in a case insensitive manner for retrieval of the user profile.

If **accountName** does not correspond to an existing user profile, the protocol server MUST send a SOAP fault message.

**colleagueAccountName:** A string that specifies the login name of the user profile of the colleague to be disassociated from the user profile specified in **accountName**. If this element does not have a value or is not specified, the protocol server MUST use the user profile of the current user. The protocol server MUST treat **colleagueAccountName** in a case insensitive manner for retrieval of the user profile.

If **colleagueAccountName** does not correspond to an existing user profile or the user profile specified by **colleagueAccountName** does not refer to a currently associated colleague, the protocol server MUST send a SOAP fault message.

#### 3.1.4.33.2.2 RemoveColleagueResponse

The **RemoveColleagueResponse** element is the body of the response **WSDL** message for the **RemoveColleague** WSDL operation.

```
<s:element name="RemoveColleagueResponse">
  <s:complexType/>
</s:element>
```

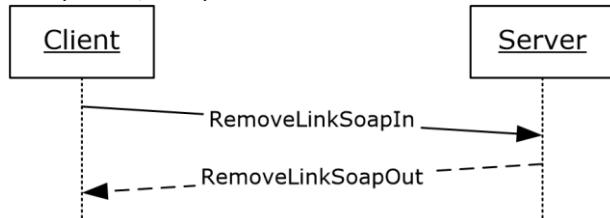
### 3.1.4.34 RemoveLink

The **RemoveLink** **WSDL** operation is used to update the specified **user profile** by deleting a specified **quick link** property present in the user profile.

```
<wsdl:operation name="RemoveLink">
  <wsdl:input message="tns:RemoveLinkSoapIn"/>
```

```
<wsdl:output message="tns:RemoveLinkSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **RemoveLinkSoapIn** request WSDL message and the protocol server responds with a **RemoveLinkSoapOut** response WSDL message. If the user is not authorized to administer the user profile service and is not updating his or her own user profile, the protocol server MUST send a **SOAP fault** message.



**Figure 36: RemoveLink operation**

### 3.1.4.34.1 Messages

The following **WSDL** message definitions are specific to this operation.

#### 3.1.4.34.1.1 RemoveLinkSoapIn

The **RemoveLinkSoapIn** message is the request **WSDL** message for the **RemoveLink** WSDL operation.

The **SOAP action** value of the message is defined as:

```
http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/RemoveLink
```

The **SOAP body** contains a **RemoveLink** element.

#### 3.1.4.34.1.2 RemoveLinkSoapOut

The **RemoveLinkSoapOut** message is the response **WSDL** message for the **RemoveLink** WSDL operation.

The **SOAP body** contains a **RemoveLinkResponse** element.

### 3.1.4.34.2 Elements

The following **XML schema** element definitions are specific to this operation.

#### 3.1.4.34.2.1 RemoveLink

The **RemoveLink** element is the body of the request **WSDL** message for the **RemoveLink** WSDL operation.

```
<s:element name="RemoveLink">
  <s:complexType>
    <s:sequence>
      <s:element name="accountName" type="s:string" minOccurs="0" maxOccurs="1"/>
      <s:element name="id" type="s:int" minOccurs="1" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**accountName:** A string that specifies the **login name** of the **user profile** to be updated. This value MUST be less than or equal to 400 characters in length. If this element is not specified or does not have a value, the protocol server MUST use the user profile of the **current user**, and return a **SOAP fault** message if the current user does not have a user profile. The protocol server MUST treat **accountName** in a case insensitive manner for retrieval of the user profile.

If **accountName** does not correspond to an existing user profile, the protocol server MUST send a SOAP fault message.

**id:** An **int** that specifies the unique identifier of the **quick link** to be removed. If this identifier does not refer to a quick link currently present in the user profile, the protocol server MUST send a SOAP fault message.

### 3.1.4.34.2 RemoveLinkResponse

The **RemoveLinkResponse** element is the body of the response **WSDL** message for the **RemoveLink** WSDL operation.

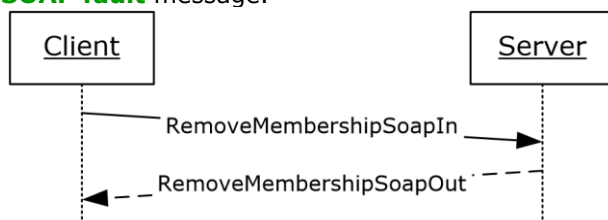
```
<s:element name="RemoveLinkResponse">
  <s:complexType/>
</s:element>
```

### 3.1.4.35 RemoveMembership

The **RemoveMembership** **WSDL** operation is used to update the specified **user profile** by deleting a specified **membership** present in the user profile.

```
<wsdl:operation name="RemoveMembership">
  <wsdl:input message="tns:RemoveMembershipSoapIn"/>
  <wsdl:output message="tns:RemoveMembershipSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **RemoveMembershipSoapIn** request WSDL message and the protocol server responds with a **RemoveMembershipSoapOut** response WSDL message. If the user is not authorized to administer the user profile service, the protocol server MUST send a **SOAP fault** message.



**Figure 37: RemoveMembership operation**

#### 3.1.4.35.1 Messages

The following **WSDL** message definitions are specific to this operation.

##### 3.1.4.35.1.1 RemoveMembershipSoapIn

The **RemoveMembershipSoapIn** message is the request **WSDL** message for the **RemoveMembership** WSDL operation.

The **SOAP action** value of the message is defined as:

```
http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/RemoveMembership
```

The **SOAP body** contains a **RemoveMembership** element.

### 3.1.4.35.1.2 RemoveMembershipSoapOut

The **RemoveMembershipSoapOut** message is the response **WSDL** message for the **RemoveMembership** WSDL operation.

The **SOAP body** contains a **RemoveMembershipResponse** element.

### 3.1.4.35.2 Elements

The following **XML schema** element definitions are specific to this operation.

#### 3.1.4.35.2.1 RemoveMembership

The **RemoveMembership** element is the body of the request **WSDL** message for the **RemoveMembership** WSDL operation.

```
<s:element name="RemoveMembership">
  <s:complexType>
    <s:sequence>
      <s:element name="accountName" type="s:string" minOccurs="0" maxOccurs="1"/>
      <s:element name="sourceInternal" type="s1:guid" minOccurs="1" maxOccurs="1"/>
      <s:element name="sourceReference" type="s:string" minOccurs="0" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**accountName:** A string that specifies the **login name** of the **user profile** to be updated. This value **MUST** be less than or equal to 400 characters in length. If this element is not specified or does not have a value, the protocol server **MUST** use the user profile of the **current user**, and return a **SOAP fault** message if the current user does not have a user profile. The protocol server **MUST** treat **accountName** in a case insensitive manner for retrieval of the user profile.

If **accountName** does not correspond to an existing user profile, the protocol server **MUST** send a SOAP fault message.

**sourceInternal:** A **guid** type that specifies the original source of members for the **member group**. The **GUID** **MUST** be one of the following values:

Value	Description
A88B9DCB-5B82-41E4-8A19-17672F307B95	The member group was originally populated from a <b>distribution list</b> .
8BB1220F-DE8B-4771-AC3A-0551242CF2BD	The member group was originally populated from a <b>site</b> .

**sourceReference:** A string that uniquely identifies the member group to be removed. If there is not an existing **membership** with the same **sourceInternal** and **sourceReference** values in the user profile specified by **accountName**, the protocol server **MUST** send a SOAP fault message.

#### 3.1.4.35.2.2 RemoveMembershipResponse

The **RemoveMembershipResponse** element is the body of the response **WSDL** message for the **RemoveMembership** WSDL operation.

```
<s:element name="RemoveMembershipResponse">
  <s:complexType/>
```

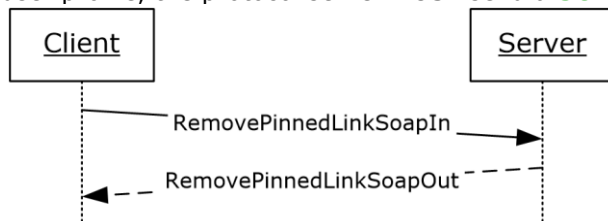
```
</s:element>
```

### 3.1.4.36 RemovePinnedLink

The **RemovePinnedLink WSDL** operation is used to update the specified **user profile** by deleting a specified **pinned link** present in the user profile.

```
<wsdl:operation name="RemovePinnedLink">  
  <wsdl:input message="tns:RemovePinnedLinkSoapIn"/>  
  <wsdl:output message="tns:RemovePinnedLinkSoapOut"/>  
</wsdl:operation>
```

The protocol client sends a **RemovePinnedLinkSoapIn** request WSDL message and the protocol server responds with a **RemovePinnedLinkSoapOut** response WSDL message. If the user is not authorized to administer the user profile service and is not updating his or her own user profile, the protocol server **MUST** send a **SOAP fault** message.



**Figure 38: RemovePinnedLink operation**

#### 3.1.4.36.1 Messages

The following **WSDL** message definitions are specific to this operation.

##### 3.1.4.36.1.1 RemovePinnedLinkSoapIn

The **RemovePinnedLinkSoapIn** message is the request **WSDL** message for the **RemovePinnedLink** WSDL operation.

The **SOAP action** value of the message is defined as:

```
http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/RemovePinnedLink
```

The **SOAP body** contains a **RemovePinnedLink** element.

##### 3.1.4.36.1.2 RemovePinnedLinkSoapOut

The **RemovePinnedLinkSoapOut** message is the response **WSDL** message for the **RemovePinnedLink** WSDL operation.

The **SOAP body** contains a **RemovePinnedLinkResponse** element.

#### 3.1.4.36.2 Elements

The following **XML schema** element definitions are specific to this operation.

##### 3.1.4.36.2.1 RemovePinnedLink

The **RemovePinnedLink** element is the body of the request **WSDL** message for the **RemovePinnedLink** WSDL operation.

```

<s:element name="RemovePinnedLink">
  <s:complexType>
    <s:sequence>
      <s:element name="accountName" type="s:string" minOccurs="0" maxOccurs="1"/>
      <s:element name="id" type="s:int" minOccurs="1" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>

```

**accountName:** A string that specifies the **login name** of the **user profile** to be updated. This value MUST be less than or equal to 400 characters in length. The protocol server MUST treat **accountName** in a case insensitive manner for retrieval of the user profile.

If **accountName** is not specified or does not have a value and there is not an existing user profile for the **current user**, then a new user profile MAY <15> be created which represents the current user and the protocol server MUST send a **SOAP fault** message.

If **accountName** is not specified or does not have a value and there is an existing user profile for the current user, then the protocol server MUST update that user profile.

If **accountName** is specified and does not match the login name of an existing user profile, the protocol server MUST send a SOAP fault message.

**id:** An **int** that specifies the unique identifier for the **pinned link** to be removed.

If **id** does not refer to a pinned link currently present in the user profile, the protocol server MUST send a SOAP fault message.

### 3.1.4.36.2 RemovePinnedLinkResponse

The **RemovePinnedLinkResponse** element is the body of the response **WSDL** message for the **RemovePinnedLink** WSDL operation.

```

<s:element name="RemovePinnedLinkResponse">
  <s:complexType/>
</s:element>

```

### 3.1.4.37 UpdateColleaguePrivacy

The **UpdateColleaguePrivacy** **WSDL** operation is used to update the specified **user profile** by updating the **privacy level** of a **colleague** association.

```

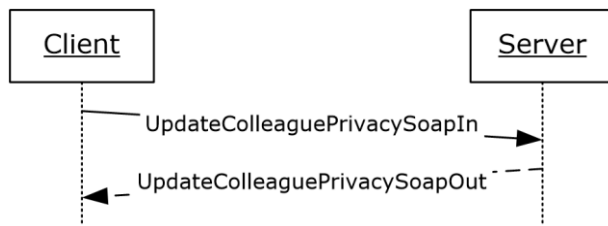
<wsdl:operation name="UpdateColleaguePrivacy">
  <wsdl:input message="tns:UpdateColleaguePrivacySoapIn"/>
  <wsdl:output message="tns:UpdateColleaguePrivacySoapOut"/>
</wsdl:operation>

```

The protocol client sends an **UpdateColleaguePrivacySoapIn** request WSDL message and the protocol server responds with an **UpdateColleaguePrivacySoapOut** response WSDL message.

If the **current user** is not authorized to administer the user profile service or is not updating his or her own user profile, the protocol server MUST send a **SOAP fault** message.

Additionally, if the user does not have the **Use Personalization Features** permission, the protocol server MUST send a SOAP fault message.



**Figure 39: UpdateColleaguePrivacy operation**

### 3.1.4.37.1 Messages

The following **WSDL** message definitions are specific to this operation.

#### 3.1.4.37.1.1 UpdateColleaguePrivacySoapIn

The **UpdateColleaguePrivacySoapIn** message is the request **WSDL** message for the **UpdateColleaguePrivacy** WSDL operation.

The **SOAP action** value of the message is defined as:

```
http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/UpdateColleaguePrivacy
```

The **SOAP body** contains an **UpdateColleaguePrivacy** element.

#### 3.1.4.37.1.2 UpdateColleaguePrivacySoapOut

The **UpdateColleaguePrivacySoapOut** message is the response **WSDL** message for the **UpdateColleaguePrivacy** WSDL operation.

The **SOAP body** contains an **UpdateColleaguePrivacyResponse** element.

### 3.1.4.37.2 Elements

The following **XML schema** element definitions are specific to this operation.

#### 3.1.4.37.2.1 UpdateColleaguePrivacy

The **UpdateColleaguePrivacy** element is the body of the request **WSDL** message for the **UpdateColleaguePrivacy** WSDL operation.

```

<s:element name="UpdateColleaguePrivacy">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="accountName" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="colleagueAccountName" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="newPrivacy" type="tns:Privacy" />
    </s:sequence>
  </s:complexType>
</s:element>
  
```

**accountName:** A string that specifies the **login name** of the **user profile** to be updated. This value MUST be less than or equal to 400 characters in length. The protocol server MUST treat **accountName** in a case insensitive manner for retrieval of the user profile.

If **accountName** is not specified or does not have a value and there is not an existing user profile for the **current user**, then a new user profile MAY [≤16>](#) be created which represents the current user and the protocol server MUST update that user profile.



If **accountName** is not specified or does not have a value and there is an existing user profile for the current user, then the protocol server MUST update that user profile.

If **accountName** is specified and does not match the login name of an existing user profile, then the protocol server MUST send a **SOAP fault** message.

**colleagueAccountName**: A string that specifies the login name of the user profile of the colleague to be updated. This string MUST be less than or equal to 400 characters in length. If this element is not specified or does not have a value, the protocol server MUST use the user profile of the current user. The user specified by **colleagueAccountName** MUST NOT be the same as the user specified by **accountName**; otherwise the protocol server MUST send a SOAP fault message. The protocol server MUST treat **colleagueAccountName** in a case insensitive manner for retrieval of the user profile.

If **colleagueAccountName** is not specified or does not have a value and there is not an existing user profile for the current user, then a new user profile MUST be created which represents the current user and the protocol server MUST send a SOAP fault message.

If **colleagueAccountName** is not specified or does not have a value and there is an existing user profile for the current user, then the protocol server MUST send a SOAP fault message.

If **colleagueAccountName** is specified and does not match the login name of an existing user profile, then the protocol server MUST send a SOAP fault message.

If **colleagueAccountName** is specified and does match the login name of an existing user profile but that user profile is not a colleague of the user with the user profile specified by **accountName**, the protocol server MUST send a SOAP fault message.

**newPrivacy**: A value that MUST correspond to a value in the **Privacy** (section [2.2.5.4](#)) enumeration. If the current **privacy level** is not set to NotSet, this value MUST NOT be set to NotSet. If this value is set to "NotSet" when the current privacy level is not set to NotSet the server MUST return a SOAP fault message. This is the updated privacy level that will be assigned to the specified **colleague** relationship.

### 3.1.4.37.2 UpdateColleaguePrivacyResponse

The **UpdateColleaguePrivacyResponse** element is the body of the response **WSDL** message for the **UpdateColleaguePrivacy** WSDL operation.

```
<s:element name="UpdateColleaguePrivacyResponse">
  <s:complexType/>
</s:element>
```

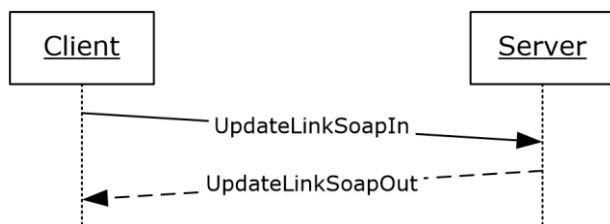
### 3.1.4.38 UpdateLink

The **UpdateLink WSDL** operation is used to update a **quick link** for the specified user.

```
<wsdl:operation name="UpdateLink">
  <wsdl:input message="tns:UpdateLinkSoapIn"/>
  <wsdl:output message="tns:UpdateLinkSoapOut"/>
</wsdl:operation>
```

The protocol client sends an **UpdateLinkSoapIn** request WSDL message and the protocol server responds with an **UpdateLinkSoapOut** response WSDL message. If the user is not authorized to administer the user profile service and is not updating his or her own **user profile**, the protocol server MUST send a **SOAP fault** message.

Additionally, if the user does not have the **Use Personalization Features** permission, the protocol server MUST send a SOAP fault message.



**Figure 40: UpdateLink operation**

### 3.1.4.38.1 Messages

The following **WSDL** message definitions are specific to this operation.

#### 3.1.4.38.1.1 UpdateLinkSoapIn

The **UpdateLinkSoapIn** message is the request **WSDL** message for the **UpdateLink** WSDL operation.

The **SOAP action** value of the message is defined as:

```
http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/UpdateLink
```

The **SOAP body** contains an **UpdateLink** element.

#### 3.1.4.38.1.2 UpdateLinkSoapOut

The **UpdateLinkSoapOut** message is the response **WSDL** message for the **UpdateLink** WSDL operation.

The **SOAP body** contains an **UpdateLinkResponse** element.

### 3.1.4.38.2 Elements

The following **XML schema** element definitions are specific to this operation.

#### 3.1.4.38.2.1 UpdateLink

The **UpdateLink** element is the body of the request **WSDL** message for the **UpdateLink** WSDL operation.

```

<s:element name="UpdateLink">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="accountName" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="data" type="tns:QuickLinkData" />
    </s:sequence>
  </s:complexType>
</s:element>
  
```

**accountName:** A string that specifies the **login name** of the **user profile** to be updated. This value **MUST** be less than or equal to 400 characters in length. The protocol server **MUST** treat **accountName** in a case insensitive manner for retrieval of the user profile.

If **accountName** is not specified or does not have a value and there is not an existing user profile for the **current user**, then a new user profile **MAY** [<17>](#) be created which represents the user and the protocol server **MUST** update that user profile.

If **accountName** is not specified or does not have a value and there is an existing user profile for the current user, then the protocol server **MUST** update that user profile.

If **accountName** is specified and does not match the login name of an existing user profile, the protocol server MUST send a **SOAP fault** message.

**data:** The **QuickLinkData** (section [2.2.4.18](#)) to be updated.

If the identifier of data does not match the identifier of an existing **quick link**, the protocol server MUST send a SOAP fault message.

### 3.1.4.38.2.2 UpdateLinkResponse

The **UpdateLinkResponse** element is the body of the response **WSDL** message for the **UpdateLink** WSDL operation.

```
<s:element name="UpdateLinkResponse">
  <s:complexType/>
</s:element>
```

### 3.1.4.39 UpdateMembershipPrivacy

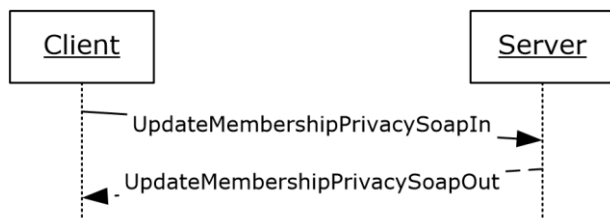
The **UpdateMembershipPrivacy** **WSDL** operation is used to update the specified **user profile** by updating the **privacy level** of a **membership**.

```
<wsdl:operation name="UpdateMembershipPrivacy">
  <wsdl:input message="tns:UpdateMembershipPrivacySoapIn"/>
  <wsdl:output message="tns:UpdateMembershipPrivacySoapOut"/>
</wsdl:operation>
```

The protocol client sends an **UpdateMembershipPrivacySoapIn** request WSDL message and the protocol server responds with an **UpdateMembershipPrivacySoapOut** response WSDL message.

If the **current user** is not authorized to administer the user profile service or is not updating his or her own user profile, the protocol server MUST send a **SOAP fault** message.

Additionally, if the user does not have the **Use Personalization Features** permission, the protocol server MUST send a SOAP fault message.



**Figure 41: UpdateMembershipPrivacy operation**

#### 3.1.4.39.1 Messages

The following **WSDL** message definitions are specific to this operation.

##### 3.1.4.39.1.1 UpdateMembershipPrivacySoapIn

The **UpdateMembershipPrivacySoapIn** message is the request **WSDL** message for the **UpdateMembershipPrivacy** WSDL operation.

The **SOAP action** value of the message is defined as:

http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/UpdateMembershipPrivacy

The **SOAP body** contains an **UpdateMembershipPrivacy** element.

### 3.1.4.39.1.2 UpdateMembershipPrivacySoapOut

The **UpdateMembershipPrivacySoapOut** message is the response **WSDL** message for the **UpdateMembershipPrivacy** WSDL operation.

The **SOAP body** contains an **UpdateMembershipPrivacyResponse** element.

### 3.1.4.39.2 Elements

The following **XML schema** element definitions are specific to this operation.

#### 3.1.4.39.2.1 UpdateMembershipPrivacy

The **UpdateMembershipPrivacy** element is the body of the request **WSDL** message for the **UpdateMembershipPrivacy** WSDL operation.

```
<s:element name="UpdateMembershipPrivacy">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="accountName" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="sourceInternal" type="s1:guid" />
      <s:element minOccurs="0" maxOccurs="1" name="sourceReference" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="newPrivacy" type="tns:Privacy" />
    </s:sequence>
  </s:complexType>
</s:element>
```

**accountName:** A string that specifies the **login name** of the **user profile** to be updated. This value MUST be less than or equal to 400 characters in length. The protocol server MUST treat **accountName** in a case insensitive manner for retrieval of the user profile.

If **accountName** is not specified or does not have a value and there is not an existing user profile for the **current user**, then a new user profile MUST be created which represents the user and the protocol server MUST update that user profile.

If **accountName** is not specified or does not have a value and there is an existing user profile for the current user, then the protocol server MUST update that user profile.

If **accountName** is specified and does not match the login name of an existing user profile, the protocol server MUST send a **SOAP fault** message.

**sourceInternal:** A **guid** type that specifies the original source of members for the **member group**. The **GUID** MUST be one of the following values:

Value	Description
A88B9DCB-5B82-41E4-8A19-17672F307B95	The member group was originally populated from a <b>distribution list</b> .
8BB1220F-DE8B-4771-AC3A-0551242CF2BD	The member group was originally populated from a SharePoint site.

**sourceReference:** A string that uniquely identifies the member group. This element is used to determine the specific **membership** to be updated.

If the specified **sourceInternal** and **sourceReference** values don't match an existing membership for the user specified by **accountName**, the protocol server MUST send a SOAP fault message.

If the member group does not exist, the server MUST return a SOAP fault message.

If the member group exists, but does not include the user specified by **accountName**, the server MUST return a SOAP fault message.

**newPrivacy**: A value that MUST correspond to a **Privacy** (section 2.2.5.4) enumeration. This value MUST NOT be NotSet. If this value is set to NotSet the server MUST return a SOAP fault message. This is the updated **privacy level** that will be assigned to the specified membership.

### 3.1.4.39.2.2 UpdateMembershipPrivacyResponse

The **UpdateMembershipPrivacyResponse** element is the body of the response **WSDL** message for the **UpdateMembershipPrivacy** WSDL operation.

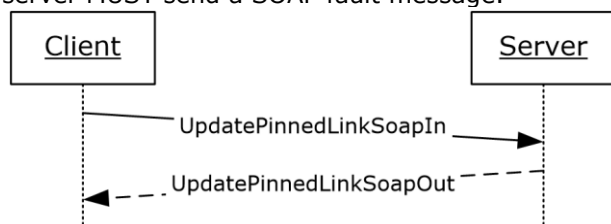
```
<s:element name="UpdateMembershipPrivacyResponse">
  <s:complexType/>
</s:element>
```

### 3.1.4.40 UpdatePinnedLink

The **UpdatePinnedLink** **WSDL** operation is used to update a **pinned link** for the specified user.

```
<wsdl:operation name="UpdatePinnedLink">
  <wsdl:input message="tns:UpdatePinnedLinkSoapIn"/>
  <wsdl:output message="tns:UpdatePinnedLinkSoapOut"/>
</wsdl:operation>
```

The protocol client sends an **UpdatePinnedLinkSoapIn** request WSDL message and the protocol server responds with an **UpdatePinnedLinkSoapOut** response WSDL message. If the user is not authorized to administer the user profile service and is not updating his or her own **user profile**, the protocol server MUST send a **SOAP fault** message. Additionally, if the user does not have the **Use Personalization Features** permission, the protocol server MUST send a SOAP fault message.



**Figure 42: UpdatePinnedLink operation**

#### 3.1.4.40.1 Messages

The following **WSDL** message definitions are specific to this operation.

##### 3.1.4.40.1.1 UpdatePinnedLinkSoapIn

The **UpdatePinnedLinkSoapIn** message is the request **WSDL** message for the **UpdatePinnedLink** WSDL operation.

The **SOAP action** value of the message is defined as:

The **SOAP body** contains an **UpdatePinnedLink** element.

#### 3.1.4.40.1.2 UpdatePinnedLinkSoapOut

The **UpdatePinnedLinkSoapOut** message is the response **WSDL** message for the **UpdatePinnedLink** WSDL operation.

The **SOAP body** contains an **UpdatePinnedLinkResponse** element.

#### 3.1.4.40.2 Elements

The following **XML schema** element definitions are specific to this operation.

##### 3.1.4.40.2.1 UpdatePinnedLink

The **UpdatePinnedLink** element is the body of the request **WSDL** message for the **UpdatePinnedLink** WSDL operation.

```
<s:element name="UpdatePinnedLink">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="accountName" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="data" type="tns:PinnedLinkData" />
    </s:sequence>
  </s:complexType>
</s:element>
```

**accountName:** A string that specifies the **login name** of the **user profile** to be updated. This value MUST be less than or equal to 400 characters in length. The protocol server MUST treat **accountName** in a case insensitive manner for retrieval of the user profile.

If **accountName** is not specified or does not have a value and there is not an existing user profile for the **current user**, then a new user profile MUST be created which represents the user and the protocol server MUST update that user profile.

If **accountName** is not specified or does not have a value and there is an existing user profile for the current user, then the protocol server MUST update that user profile.

If **accountName** is specified and does not match the login name of an existing user profile, the protocol server MUST send a **SOAP fault** message.

**data:** The **PinnedLinkData** (section [2.2.4.15](#)) to be updated.

If the identifier of data does not match the identifier of an existing **pinned link**, the protocol server MUST send a SOAP fault message.

##### 3.1.4.40.2.2 UpdatePinnedLinkResponse

The **UpdatePinnedLinkResponse** element is the body of the response **WSDL** message for the **UpdatePinnedLink** WSDL operation.

```
<s:element name="UpdatePinnedLinkResponse">
  <s:complexType/>
</s:element>
```

### **3.1.5 Timer Events**

None.

### **3.1.6 Other Local Events**

None.

## 4 Protocol Examples

The following sections show examples of protocol requests and responses for various scenarios around the user profile web service.

### 4.1 Create User Profile Example

This section demonstrates how to create a user profile using a known account name. The appropriate request is demonstrated and the expected feedback is also shown.

#### 4.1.1 Create the Profile

This protocol can be used to create **user profiles**. The protocol client calls **CreateUserProfileByAccountName** with the name of a specified user. **CreateUserProfileByAccountName** will return a set of the properties it created with the default values.

##### 4.1.1.1 Request

The following request message is constructed:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>

    <CreateUserProfileByAccountName
      xmlns="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService">
      <accountName>Contoso\Weber</accountName>
    </CreateUserProfileByAccountName>

  </soap:Body>
</soap:Envelope>
```

##### 4.1.1.2 Response

Assuming there are only two properties defined for a **user profile**, **Name** and **Address**, the following response message is returned:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <CreateUserProfileByAccountNameResponse
      xmlns="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService">
      <CreateUserProfileByAccountNameResult>
        <PropertyData>
          <IsPrivacyChanged>False</IsPrivacyChanged>
          <IsValueChanged>False</IsValueChanged>
          <Name>Name</Name>
          <Privacy>NotSet</Privacy>
          <Values>
            <ValueData>
              <Value xsi:type="xsd:string">Martin Weber</Value>
            </ValueData>
          </Values>
        </CreateUserProfileByAccountNameResult>
      </CreateUserProfileByAccountNameResponse>
    </soap:Body>
  </soap:Envelope>
```



```

    </PropertyData>
  <PropertyData>
    <IsPrivacyChanged>False</IsPrivacyChanged>
    <IsValueChanged>False</IsValueChanged>
    <Name>Address</Name>
    <Privacy>NotSet</Privacy>
    <Values>
      <ValueData>
        <Value xsi:type="xsd:string"></Value>
      </ValueData>
    </Values>
  </PropertyData>
</CreateUserProfileByAccountNameResult>
</CreateUserProfileByAccountNameResponse>
</soap:Body>
</soap:Envelope>

```

## 4.2 Commonalities Example

This example demonstrates how to separately determine the common manager, common memberships and common colleagues between two valid user accounts. Then the example will show how to use the **GetInCommon** method to determine all three commonalities at once.

### 4.2.1 GetCommonManager

The protocol client can retrieve the common manager between two users. The **GetCommonManager WSDL** operation will retrieve the common manager of the **current user** and a specified user.

#### 4.2.1.1 Request

The following request message is constructed:

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetCommonManager
xmlns="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService">
      <accountName>Contoso\Weber</accountName>
    </GetCommonManager>
  </soap:Body>
</soap:Envelope>

```

#### 4.2.1.2 Response

The protocol server returns the following response message:

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetCommonManagerResponse
xmlns="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService">
      <GetCommonManagerResult>
        <AccountName>Contoso\Coreets</AccountName>
        <Privacy>NotSet</Privacy>
        <Name>Eva Corets</Name>
        <IsInWorkGroup>false</IsInWorkGroup>
      </GetCommonManagerResult>
    </GetCommonManagerResponse>
  </soap:Body>
</soap:Envelope>

```

```

    <Email>Eva@contoso.com</Email>
    <Title>Manager</Title>
    <Url>http://www.example.com/mysite/Person.aspx?accountname=Jane.Doe</Url>
    <UserProfileID>25fa36d0-964e-48ea-8568-304bc1847f32</UserProfileID>
    <ID>0</ID>
  </GetCommonManagerResult>
</GetCommonManagerResponse>
</soap:Body>
</soap:Envelope>

```

## 4.2.2 GetCommonMemberships

The protocol client retrieves the common **memberships** between two users. The **GetCommonMemberships WSDL** operation retrieves the common memberships of the **current user** and a specified user. In this example, assume that the currently logged on user and Martin Weber both have the memberships "Some Group" and "Another Group".

### 4.2.2.1 Request

The following request message is constructed:

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetCommonMemberships
xmlns="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService"
    <accountName>Contoso\Weber</accountName>
  </GetCommonMemberships>
  </soap:Body>
</soap:Envelope>

```

### 4.2.2.2 Response

The protocol server returns the following response message:

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetCommonMembershipsResponse
xmlns="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService">
      <GetCommonMembershipsResult>
        <MembershipData>
          <Source>DistributionList</Source>
          <MemberGroup>
            <SourceInternal>25fa36d0-964e-48ea-8568-304bc1847f37</SourceInternal>
            <SourceReference>string</SourceReference>
          </MemberGroup>
          <DisplayName>Some Group</DisplayName>
          <Privacy>NotSet</Privacy>
          <MailNickname>Some Group</MailNickname>
          <Url>mailto:somegroup@contoso.com</Url>
          <ID>1</ID>
          <MemberGroupID>1</MemberGroupID>
        </MembershipData>
        <MembershipData>
          <Source>DistributionList</Source>
          <MemberGroup>

```

```

        <SourceInternal>25fa36d0-964e-48ea-8568-304bc1847f33</SourceInternal>
        <SourceReference>string</SourceReference>
    </MemberGroup>
    <DisplayName>Another Group</DisplayName>
    <Privacy>Contacts</Privacy>
    <MailNickname>Another Group</MailNickname>
    <Url>mailto:anothergroup@contoso.com</Url>
    <ID>4</ID>
    <MemberGroupID>5</MemberGroupID>
</MembershipData>
</GetCommonMembershipsResult>
</GetCommonMembershipsResponse>
</soap:Body>
</soap:Envelope>

```

### 4.2.3 GetCommonColleagues

A protocol client retrieves the common colleagues between two users. The **GetCommonColleagues** WSDL operation will retrieve the common **colleagues** of the **current user** and a specified user. In this example assume the current user and Martin Weber are both colleagues with Cassie Hicks and John Glen."

#### 4.2.3.1 Request

The following request message is constructed:

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetCommonColleagues
xmlns="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService">
      <accountName>Contoso\Weber</accountName>
    </GetCommonColleagues>
  </soap:Body>
</soap:Envelope>

```

#### 4.2.3.2 Response

The protocol server returns the following response message:

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetCommonColleaguesResponse
xmlns="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService">
      <GetCommonColleaguesResult>
        <ContactData>
          <AccountName>Contoso\Hicks</AccountName>
          <Privacy>NotSet</Privacy>
          <Name>Cassie Hicks</Name>
          <IsInWorkGroup>>false</IsInWorkGroup>
          <Email>Hicks@contoso.com</Email>
          <Title>Developer</Title>
          <Url>http://www.conotoso.com/mysite/Person.aspx?accountname=Contoso\Hicks</Url>
          <UserProfileID>25fa36d0-964e-48ea-8568-304bc1847f39</UserProfileID>
          <ID>0</ID>
        </ContactData>
      </GetCommonColleaguesResult>
    </GetCommonColleaguesResponse>
  </soap:Body>
</soap:Envelope>

```

```

    <ContactData>
      <AccountName>Contoso\Glen</AccountName>
      <Privacy>NotSet</Privacy>
      <Name>John Glen</Name>
      <IsInWorkGroup>>false</IsInWorkGroup>
      <Email>Glen@contoso.com</Email>
      <Title>Tester</Title>
    <Url>http://www.conotoso.com/mysite/Person.aspx?accountname=Contoso\Glen</Url>
      <UserProfileID>25fa36d0-964e-48ea-8568-304bc1847f3a</UserProfileID>
      <ID>0</ID>
    </ContactData>
  </GetCommonColleaguesResult>
</GetCommonColleaguesResponse>
</soap:Body>
</soap:Envelope>

```

## 4.2.4 GetInCommon

A protocol client retrieves the set of common manager, **memberships** and **colleagues** between the **current user** and another user. The protocol client could call the **GetCommonManager**, **GetCommonMemberships**, and **GetCommonColleagues WSDL** operations or it could call the **GetInCommon** WSDL operation to retrieve similar data for all three in the set at once.

### 4.2.4.1 Request

The following request message is constructed:

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetInCommon
      xmlns="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService">
      <accountName>Contoso\Weber</accountName>
    </GetInCommon>
  </soap:Body>
</soap:Envelope>

```

### 4.2.4.2 Response

The protocol server returns the following response message:

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetInCommonResponse
      xmlns="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService">
      <GetInCommonResult>
        <Manager>
          <AccountName>Contoso\Corets</AccountName>
          <Privacy>NotSet</Privacy>
          <Name>Eva Corets</Name>
          <IsInWorkGroup>>false</IsInWorkGroup>
          <Email>Eva@contoso.com</Email>
          <Title>Manager</Title>
        <Url>http://www.conotoso.com/mysite/Person.aspx?accountname=Contoso\Corets</Url>
          <UserProfileID>25fa36d0-964e-48ea-8568-304bc1847f32</UserProfileID>
          <ID>0</ID>
        </Manager>
      </GetInCommonResult>
    </GetInCommonResponse>
  </soap:Body>
</soap:Envelope>

```

```

    </Manager>
    <Colleagues>
      <ContactData>
        <AccountName>Contoso\Hicks</AccountName>
        <Privacy>NotSet</Privacy>
        <Name>Cassie Hicks</Name>
        <IsInWorkGroup>>false</IsInWorkGroup>
        <Email>Hicks@contoso.com</Email>
        <Title>Developer</Title>
      <Url>http://www.conotoso.com/mysite/Person.aspx?accountname=Contoso\Hicks</Url>
        <UserProfileID>25fa36d0-964e-48ea-8568-304bc1847f39</UserProfileID>
        <ID>0</ID>
      </ContactData>
      <ContactData>
        <AccountName>Contoso\Glen</AccountName>
        <Privacy>NotSet</Privacy>
        <Name>John Glen</Name>
        <IsInWorkGroup>>false</IsInWorkGroup>
        <Email>Glen@contoso.com</Email>
        <Title>Tester</Title>
      <Url>http://www.conotoso.com/mysite/Person.aspx?accountname=Contoso\Glen</Url>
        <UserProfileID>25fa36d0-964e-48ea-8568-304bc1847f3a</UserProfileID>
        <ID>0</ID>
      </ContactData>
    </Colleagues>
    <Memberships>
      <MembershipData>
        <Source>DistributionList</Source>
        <MemberGroup>
          <SourceInternal>25fa36d0-964e-48ea-8568-304bc1847f37</SourceInternal>
          <SourceReference>string</SourceReference>
        </MemberGroup>
        <DisplayName>Some Group</DisplayName>
        <Privacy>NotSet</Privacy>
        <MailNickname>Some Group</MailNickname>
        <Url>mailto:somegroup@contoso.com</Url>
        <ID>1</ID>
        <MemberGroupID>1</MemberGroupID>
      </MembershipData>
      <MembershipData>
        <Source>DistributionList</Source>
        <MemberGroup>
          <SourceInternal>25fa36d0-964e-48ea-8568-304bc1847f33</SourceInternal>
          <SourceReference>string</SourceReference>
        </MemberGroup>
        <DisplayName>Another Group</DisplayName>
        <Privacy>Contacts</Privacy>
        <MailNickname>Another Group</MailNickname>
        <Url>mailto:anothergroup@contoso.com</Url>
        <ID>4</ID>
        <MemberGroupID>5</MemberGroupID>
      </MembershipData>
    </Memberships>
  </GetInCommonResult>
</GetInCommonResponse>
</soap:Body>
</soap:Envelope>

```

### 4.3 GetUserProfile Example

This example demonstrates a common method for retrieving a user profile if the name of the user is known. The first step demonstrated is the use of the GetUserProfileSchema WSDL operation so that the client can understand the format of the profile. Second, the GetUserProfileByName WSDL operation is demonstrated. A request message for a single user profile is sent to the server, and then a GetUserProfileByNameResponse message, matching the schema retrieved is returned, which contain the user profile data. Error scenarios are not addressed.

### 4.3.1 Retrieve Profile Definition

A **user profile** is a set of properties that can change. For a protocol client to understand the information retrieved from the **GetUserProfileByName WSDL** operation, it first calls the **GetUserProfileSchema WSDL** operation.

#### 4.3.1.1 Obtain the Schema

When a protocol client interacts with **user profiles**, it calls the **GetUserProfileSchema WSDL** operation to obtain a list of the various properties in the profile.

##### 4.3.1.1.1 Request

The following request message is constructed:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetUserProfileSchema
xmlns="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService" />
  </soap:Body>
</soap:Envelope>
```

##### 4.3.1.1.2 Response

Assume that there are only two properties defined for a **user profile**. The protocol server responds with the following response message:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetUserProfileSchemaResponse
xmlns="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService">
      <GetUserProfileSchemaResult>
        <PropertyInfo>
          <Name>Name</Name>
          <DisplayOrder>1</DisplayOrder>          <MaximumShown>1</MaximumShown>
          <IsAdminEditable>True</IsAdminEditable>          <IsSearchable>True</IsSearchable>
        </PropertyInfo>
        <IsSystem>True</IsSystem>
        <DisplayName>Name</DisplayName>
        <Type>String</Type>
        <AllowPolicyOverride>False</AllowPolicyOverride>
        <DefaultPrivacy>NotSet</DefaultPrivacy>
        <IsAlias>False</IsAlias>
        <IsColleagueEventLog>False</IsColleagueEventLog>
        <IsRequired>True</IsRequired>
        <IsUserEditable>True</IsUserEditable>
        <IsVisibleOnEditor>True</IsVisibleOnEditor>
        <IsVisibleOnViewer>True</IsVisibleOnViewer>
        <IsReplicable>True</IsReplicable>
        <UserOverridePrivacy>False</UserOverridePrivacy>
        <Length>400</Length>
        <IsImported>True</IsImported>
        <IsMultiValue>False</IsMultiValue>
        <ChoiceType>None</ChoiceType>          <TermSetId>FC2C18F4-832B-4b87-87C5-
D0D60E9AEFD1</TermSetId>
      </PropertyInfo>
    </PropertyInfo>
  </soap:Body>
</soap:Envelope>
```

```

        <Name>Address</Name>                <DisplayOrder>2</DisplayOrder>
<MaximumShown>1</MaximumShown>
    <IsAdminEditable>True</IsAdminEditable>    <IsSearchable>True</IsSearchable>
<IsSystem>True</IsSystem>
    <DisplayName>User's Address</DisplayName>
    <Type>String</Type>
    <AllowPolicyOverride>False</AllowPolicyOverride>
    <DefaultPrivacy>NotSet</DefaultPrivacy>
    <IsAlias>False</IsAlias>
    <IsColleagueEventLog>False</IsColleagueEventLog>
    <IsRequired>True</IsRequired>
    <IsUserEditable>True</IsUserEditable>
    <IsVisibleOnEditor>True</IsVisibleOnEditor>
    <IsVisibleOnViewer>True</IsVisibleOnViewer>
    <IsReplicable>True</IsReplicable>
    <UserOverridePrivacy>False</UserOverridePrivacy>
    <Length>1000</Length>
    <IsImported>True</IsImported>
    <IsMultiValue>False</IsMultiValue>
    <ChoiceType>None</ChoiceType>          <TermSetId>FC60F505-297D-4d6a-B7BE-
0EFF215F17C6</TermSetId>
    </PropertyInfo>
</GetUserProfileSchemaResult>
</GetUserProfileSchemaResponse>
</soap:Body>
</soap:Envelope>

```

#### 4.3.1.2 Obtain the Profile of the Specified User

Now that the schema is known, the protocol client can call the **GetUserProfileByName WSDL** operation for the **user profile** of Martin Weber.

##### 4.3.1.2.1 Request

The following request message is constructed:

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetUserProfileByName
xmlns="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService">
      <AccountName>Contoso\Weber</AccountName>
    </GetUserProfileByName>
  </soap:Body>
</soap:Envelope>

```

##### 4.3.1.2.2 Response

The protocol server returns the following response message:

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetUserProfileByNameResponse
xmlns="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService">
      <GetUserProfileByNameResult>
        <PropertyData>
          <IsPrivacyChanged>False</IsPrivacyChanged>

```

```

    <IsValueChanged>False</IsValueChanged>
    <Name>Name</Name>
    <Privacy>NotSet</Privacy>
    <Values>
      <ValueData>
        <Value xsi:type="xsd:string">Martin Weber</Value>
      </ValueData>
    </Values>
  </PropertyData>
</PropertyData>
  <IsPrivacyChanged>False</IsPrivacyChanged>
  <IsValueChanged>False</IsValueChanged>
  <Name>Address</Name>
  <Privacy>NotSet</Privacy>
  <Values>
    <ValueData>
      <Value xsi:type="xsd:string">100 Somewhere Street</Value>
    </ValueData>
  </Values>
</PropertyData>
</GetUserProfileByNameResult>
</GetUserProfileByNameResponse>
</soap:Body>
</soap:Envelope>

```

## 4.4 GetUserProfileByIndex Example

In this example, the mechanism for enumerating all **user profiles** using the **GetUserProfileByIndex WSDL** operation (section [3.1.4.23](#)) is demonstrated. Two request messages for user profiles are made in sequential order starting with the first (index 0). The server then returns the first and then the second responses in the form of **GetUserProfileByIndexResponse**, which contain the user profile data.

### 4.4.1 Enumerate Users

To retrieve all of the **user profiles** in the system, the **GetUserProfileByIndex WSDL** operation can be used. The **GetUserProfileByIndex** WSDL operation will allow the protocol client to pass in the index of the user profile to obtain, and it will return information about that user profile and the index of the next user profile if such a user profile exists.

#### 4.4.1.1 Obtain User Profiles

A protocol client calls the **GetUserProfileByIndex WSDL** operation with an index of zero to begin. The result will have a parameter called *NextValue* that will indicate the index of the next **user profile** or -1 if there are no more user profiles. In this manner a protocol client can enumerate all of the user profiles in the system.

##### 4.4.1.1.1 Request

The following request message is for the first **user profile**:

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetUserProfileByIndex
      xmlns="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService">
      <index>0</index>
    </GetUserProfileByIndex>
  </soap:Body>

```



```
</soap:Envelope>
```

The following request message is for the next user profile (assuming there is more than one user profile):

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetUserProfileByIndex
xmlns="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService">
      <index>1</index>
    </GetUserProfileByIndex>
  </soap:Body>
</soap:Envelope>
```

#### 4.4.1.1.2 Response

The protocol server returns the following response message for the first **user profile**:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetUserProfileByIndexResponse
xmlns="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService">
      <GetUserProfileByIndexResult>
        <NextValue>1</NextValue>
        <UserProfile>
          <PropertyData>
            <PropertyData>
              <IsPrivacyChanged>False</IsPrivacyChanged>
              <IsValueChanged>False</IsValueChanged>
              <Name>Name</Name>
              <Privacy>NotSet</Privacy>
              <Values>
                <ValueData>
                  <Value xsi:type="xsd:string">Martin Weber</Value>
                </ValueData>
              </Values>
            </PropertyData>
          </PropertyData>
          <PropertyData>
            <IsPrivacyChanged>False</IsPrivacyChanged>
            <IsValueChanged>False</IsValueChanged>
            <Name>SomeOtherProperty</Name>
            <Privacy>NotSet</Privacy>
            <Values>
              <ValueData>
                <Value xsi:type="xsd:string">SomeOtherValue</Value>
              </ValueData>
            </Values>
          </PropertyData>
        </UserProfile>
        <Colleagues>
          <ContactData>
            <AccountName>Contoso\Hicks</AccountName>
            <Privacy>NotSet</Privacy>
            <Name>Cassie Hicks</Name>
            <IsInWorkGroup>false</IsInWorkGroup>
            <Email>Kim@contoso.com</Email>
            <Title>Developer</Title>
            <Url></Url>
            <UserProfileID>guid</UserProfileID>
            <ID></ID>
          </ContactData>
        </Colleagues>
      </GetUserProfileByIndexResult>
    </GetUserProfileByIndexResponse>
  </soap:Body>
</soap:Envelope>
```

```

</ContactData>
<ContactData>
  <AccountName>Contoso\Glen</AccountName>
  <Privacy>NotSet</Privacy>
  <Name>John Glen</Name>
  <IsInWorkGroup>>false</IsInWorkGroup>
  <Email>Glen@contoso.com</Email>
  <Title>Tester</Title>
  <Url></Url>
  <UserProfileID>guid</UserProfileID>
  <ID></ID>
</ContactData>
</Colleagues>
<QuickLinks>
  <QuickLinkData>
    <Name>A Person</Name>
    <Group>A group</Group>
    <Privacy>NotSet</Privacy>
    <Url>http://contoso/APerson</Url>
    <ID>1</ID>
  </QuickLinkData>
  <QuickLinkData>
    <Name>Another Person</Name>
    <Group>A group</Group>
    <Privacy>NotSet</Privacy>
    <Url>http://contoso/AnotherPerson</Url>
    <ID>2</ID>
  </QuickLinkData>
</QuickLinks>
<PinnedLinks>
  <PinnedLinkData>
    <Name>Some Person</Name>
    <Url>http://contoso/SomePerson</Url>
    <ID>3</ID>
  </PinnedLinkData>
  <PinnedLinkData>
    <Name>Some Other Person</Name>
    <Url>http://contoso/SomeOtherPerson</Url>
    <ID>4</ID>
  </PinnedLinkData>
</PinnedLinks>
<Memberships>
  <MembershipData>
    <Source>SharePointSite</Source>
    <MemberGroup>
      <SourceInternal>8BB1220F-DE8B-4771-AC3A-0551242CF2BD</SourceInternal>
      <SourceReference>string</SourceReference>
    </MemberGroup>
    <Group>A group</Group>
    <DisplayName>A group</DisplayName>
    <Privacy>NotSet</Privacy>
    <MailNickname>AGroup</MailNickname>
    <Url></Url>
    <ID>10</ID>
    <MemberGroupID>1</MemberGroupID>
  </MembershipData>
  <MembershipData>
    <Source>DistributionList</Source>
    <MemberGroup>
      <SourceInternal>A88B9DCB-5B82-41E4-8A19-17672F307B95</SourceInternal>
      <SourceReference>string</SourceReference>
    </MemberGroup>
    <Group>Another group</Group>
    <DisplayName>Another group</DisplayName>
    <Privacy>NotSet</Privacy>
    <MailNickname>AnotherGroup</MailNickname>
    <Url></Url>
    <ID>11</ID>
    <MemberGroupID>1</MemberGroupID>

```

```

        </MembershipData>
    </Memberships>
</GetUserProfileByIndexResult>
</GetUserProfileByIndexResponse>
</soap:Body>
</soap:Envelope>

```

Taking note of the *NextValue* field in the response shown previously, it is not -1; thus, the second response message returned is for the second user profile:

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetUserProfileByIndexResponse
xmlns="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService">
      <GetUserProfileByIndexResult>
        <NextValue>3</NextValue>
        <UserProfile>
          <PropertyData>
            <IsPrivacyChanged>False</IsPrivacyChanged>
            <IsValueChanged>False</IsValueChanged>
            <Name>Name2</Name>
            <Privacy>NotSet</Privacy>
            <Values>
              <ValueData>
                <Value xsi:type="xsd:string">Eva Corets</Value>
              </ValueData>
            </Values>
          </PropertyData>
          <PropertyData>
            <IsPrivacyChanged>False</IsPrivacyChanged>
            <IsValueChanged>False</IsValueChanged>
            <Name>PropA</Name>
            <Privacy>NotSet</Privacy>
            <Values>
              <ValueData>
                <Value xsi:type="xsd:string">SomeOtherValue</Value>
              </ValueData>
            </Values>
          </PropertyData>
        </UserProfile>
        <Colleagues>
          <ContactData>
            <AccountName>Contoso\Counts</AccountName>
            <Privacy>NotSet</Privacy>
            <Name>Robin Counts</Name>
            <IsInWorkGroup>false</IsInWorkGroup>
            <Email>Counts@contoso.com</Email>
            <Title>Developer</Title>
            <Url></Url>
            <UserProfileID>2C3B6339-3352-47e3-8C91-1EF42934D68C</UserProfileID>
            <ID></ID>
          </ContactData>
        </Colleagues>
        <QuickLinks>
          <QuickLinkData>
            <Name>PersonABC</Name>
            <Group>GroupABC</Group>
            <Privacy>NotSet</Privacy>
            <Url>http://contoso/PersonABC</Url>
            <ID>1</ID>
          </QuickLinkData>
          <QuickLinkData>
            <Name>Another Person</Name>
            <Group>A group</Group>

```

```

        <Privacy>NotSet</Privacy>
        <Url>http://contoso/AnotherPerson</Url>
        <ID>2</ID>
    </QuickLinkData>
</QuickLinks>
<PinnedLinks>
    <PinnedLinkData>
        <Name>Person Somebody</Name>
        <Url>http://contoso/PersonSomebody</Url>
        <ID>3</ID>
    </PinnedLinkData>
    <PinnedLinkData>
        <Name>Some Other Person</Name>
        <Url>http://contoso/SomeOtherPerson</Url>
        <ID>4</ID>
    </PinnedLinkData>
</PinnedLinks>
<Memberships>
    <MembershipData>
        <Source>SharePointSite</Source>
        <MemberGroup>
            <SourceInternal>8BB1220F-DE8B-4771-AC3A-0551242CF2BD</SourceInternal>
            <SourceReference>string</SourceReference>
        </MemberGroup>
        <Group>A group</Group>
        <DisplayName>A group</DisplayName>
        <Privacy>NotSet</Privacy>
        <MailNickname>AGroup</MailNickname>
        <Url></Url>
        <ID>10</ID>
        <MemberGroupID>1</MemberGroupID>
    </MembershipData>
    <MembershipData>
        <Source>DistributionList</Source>
        <MemberGroup>
            <SourceInternal>A88B9DCB-5B82-41E4-8A19-17672F307B95</SourceInternal>
            <SourceReference>string</SourceReference>
        </MemberGroup>
        <Group>Another group</Group>
        <DisplayName>Another group</DisplayName>
        <Privacy>NotSet</Privacy>
        <MailNickname>AnotherGroup</MailNickname>
        <Url></Url>
        <ID>11</ID>
        <MemberGroupID>1</MemberGroupID>
    </MembershipData>
</Memberships>
</GetUserProfileByIndexResult>
</GetUserProfileByIndexResponse>
</soap:Body>
</soap:Envelope>

```

## 4.5 Memberships

This example shows how to synchronize the **memberships** of two users by copying memberships from the second to the first and deleting any on the first which the second does not have. The appropriate request messages and response messages are demonstrated.

### 4.5.1 Example Data

In these synchronization examples, the data begins organized in the following arrangement:

The user profile store contains two user profiles (P1, P2) representing two different directory accounts:

- P1 – Martin Weber

- P2 – Cassie Hicks

There are three member groups:

- MG1 – Marketing
- MG2 – Shipping
- MG3 – Content Management

The users have the following memberships:

- P1 – Martin Weber
  - MG1 - Marketing
  - MG3 - Content Management
  - MG4 - Sales
- P2 – Cassie Hicks
  - MG2 – Shipping
  - MG3 – Content Management

## 4.5.2 Synchronize Two Users

Assume that a **site collection administrator** requests to write a small application to copy **memberships** associated with one user to another user. The application will also remove any memberships that the second user does not have. For the data described previously, the application will copy P2's memberships to P1 using the following steps:

1. Obtain memberships for both users.
2. Add any memberships that exist in P2 to P1.
3. Remove any memberships that exist in P1 that do not exist in P2.

### 4.5.2.1 Obtain Memberships

The **GetUserMemberships WSDL** operation will return a set of **MembershipData** items for the requested user. Assume that the user is authorized to administer the user profile service so that the calls complete successfully. To get the **MembershipData** set, the **GetUserMemberships** WSDL operation is called twice; once with the first user (P1 – Martin Weber) and again with the second user (P2 – Cassie Hicks).

#### 4.5.2.1.1 Requests

The following request message for Martin Weber is constructed:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetUserMemberships
xmlns="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService">
      <accountName>Contoso\Weber</accountName>
    </GetUserMemberships>
  </soap:Body>
</soap:Envelope>
```

The following request for Cassie Hicks is constructed:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetUserMemberships
xmlns="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService">
      <accountName>Contoso\Hicks</accountName>
    </GetUserMemberships>
  </soap:Body>
</soap:Envelope>
```

#### 4.5.2.1.2 Responses

The protocol server returns the following response for Martin Weber:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetUserMembershipsResponse
xmlns="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService">
      <GetUserMembershipsResult>
        <MembershipData>
          <Source>SharePointSite</Source>
          <MemberGroup>
            <SourceInternal>8BB1220F-DE8B-4771-AC3A-0551242CF2BD</SourceInternal>
            <SourceReference>string</SourceReference>
          </MemberGroup>
          <Group>Marketing</Group>
          <DisplayName>Marketing</DisplayName>
          <Privacy>Public</Privacy>
          <MailNickname>Marketing</MailNickname>
          <Url></Url>
          <ID>1</ID>
          <MemberGroupID>1</MemberGroupID>
        </MembershipData>
        <MembershipData>
          <Source> DistributionList</Source>
          <MemberGroup>
            <SourceInternal> A88B9DCB-5B82-41E4-8A19-17672F307B95</SourceInternal>
            <SourceReference>string</SourceReference>
          </MemberGroup>
          <Group>Content Management</Group>
          <DisplayName>Content Management</DisplayName>
          <Privacy>Public</Privacy>
          <MailNickname>Content Management</MailNickname>
          <Url></Url>
          <ID>2</ID>
          <MemberGroupID>2</MemberGroupID>
        </MembershipData>
      </GetUserMembershipsResult>
    </GetUserMembershipsResponse>
  </soap:Body>
</soap:Envelope>
```

The protocol server returns the following response for Cassie Hicks:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
```

```

<soap:Body>
  <GetUserMembershipsResponse
    xmlns="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService">
    <GetUserMembershipsResult>
      <MembershipData>
        <Source>SharePointSite</Source>
        <MemberGroup>
          <SourceInternal>8BB1220F-DE8B-4771-AC3A-0551242CF2BD</SourceInternal>
          <SourceReference>string</SourceReference>
        </MemberGroup>
        <Group>Shipping</Group>
        <DisplayName>Shipping</DisplayName>
        <Privacy>Public</Privacy>
        <MailNickname>Shipping</MailNickname>
        <Url></Url>
        <ID>3</ID>
        <MemberGroupID>3</MemberGroupID>
      </MembershipData>
      <MembershipData>
        <Source>DistributionList</Source>
        <MemberGroup>
          <SourceInternal>A88B9DCB-5B82-41E4-8A19-17672F307B95</SourceInternal>
          <SourceReference>string</SourceReference>
        </MemberGroup>
        <Group>Content Management</Group>
        <DisplayName>Content Management</DisplayName>
        <Privacy>Public </Privacy>
        <MailNickname>Content Management</MailNickname>
        <Url></Url>
        <ID>2</ID>
        <MemberGroupID>2</MemberGroupID>
      </MembershipData>
    </GetUserMembershipsResult>
  </GetUserMembershipsResponse>
</soap:Body>
</soap:Envelope>

```

#### 4.5.2.2 Adding Memberships to P1 – Martin Weber

Now that the protocol client has two arrays of **memberships**, it can iterate through the second (P2 – Cassie Hicks) to find memberships that do not exist in P1 – Martin Weber. Once it finds a membership that does not exist, it calls the **AddMembership WSDL** operation. In this example, the protocol client will only find one such membership and thus call AddMembership only once: only MG2 – Shipping does not exist in P1's memberships. The **AddMembership WSDL** operation takes four parameters:

- *accountName*: Set to the name of P1 – Martin Weber, for this scenario.
- *membershipInfo*: Set to the **MembershipData** object acquired in section [4.5.2.1.2](#) by obtaining memberships.
- *group*: Set to the group defined on the **MembershipData** object that will be added.
- *privacy*: Set to the privacy level defined on the **MembershipData** object that will be added.

Note that the *group* and *privacy* parameters could be set to other values depending on the scenario.

##### 4.5.2.2.1 Requests

The following request message is constructed:

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>

```

```

    <AddMembership
xmlns="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService">
  <accountName>Contoso\Weber</accountName>
  <membershipInfo>
    <Source>SharePointSite</Source>
    <MemberGroup>
      <SourceInternal>8BB1220F-DE8B-4771-AC3A-0551242CF2BD</SourceInternal>
      <SourceReference>string</SourceReference>
    </MemberGroup>
    <Group>Shipping</Group>
    <DisplayName>Shipping</DisplayName>
    <Privacy>Public</Privacy>
    <MailNickname>Shipping</MailNickname>
    <Url>http://www.contoso.com/Shipping</Url>
    <ID>3</ID>
    <MemberGroupID>3</MemberGroupID>
  </membershipInfo>
  <group>Shipping</group>
  <privacy>Public</privacy>
</AddMembership>
</soap:Body>
</soap:Envelope>

```

#### 4.5.2.2.2 Responses

The protocol server returns the following response message:

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <AddMembershipResponse
xmlns="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService">
      <AddMembershipResult>
        <Source>SharePointSite</Source>
        <MemberGroup>
          <SourceInternal>8BB1220F-DE8B-4771-AC3A-0551242CF2BD</SourceInternal>
          <SourceReference>string</SourceReference>
        </MemberGroup>
        <Group>Shipping</Group>
        <DisplayName>Shipping</DisplayName>
        <Privacy>Public</Privacy>
        <MailNickname>Shipping</MailNickname>
        <Url>http://www.contoso.com/Shipping</Url>
        <ID>3</ID>
        <MemberGroupID>3</MemberGroupID>
      </AddMembershipResult>
    </AddMembershipResponse>
  </soap:Body>
</soap:Envelope>

```

#### 4.5.2.3 Deleting Memberships from P1 – Martin Weber

With another simple iteration the application can determine that MG1 – Marketing and MG4 – Sales needs to be removed from P1- Martin Weber. To remove these **memberships** the application will call the **RemoveMembership WSDL** operation with the following parameters:

- *accountName*: Set to the account name, which is P1 – Martin Weber.
- *sourceInternal*: Set to the value obtained from the **MemberGroupData** (section [2.2.4.12](#)) element which is part of the **MembershipData** ([2.2.4.13](#)) element.
- *sourceReference*: Set to the value obtained from the **MemberGroupData** element which is part of the **MembershipData** element.



When removing MG1 – Marketing, the application will call the **RemoveMembership** WSDL operation once with the data relevant to MG1 – Marketing.

#### 4.5.2.3.1 Requests

The following request message is constructed:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <RemoveMembership
xmlns="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService">
      <accountName>Contoso\Weber</accountName>
      <sourceInternal>8BB1220F-DE8B-4771-AC3A-0551242CF2BD</sourceInternal>
      <sourceReference>Insert string here </sourceReference>
    </RemoveMembership>
  </soap:Body>
</soap:Envelope>
```

#### 4.5.2.3.2 Responses

The protocol server returns the following response message:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <RemoveMembershipResponse
xmlns="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService" />
  </soap:Body>
</soap:Envelope>
```

## **5 Security**

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

None.

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

None.

## 6 Appendix A: Full WSDL

For ease of implementation, the full **WSDL** is provided as follows.

```
<?xml version="1.0" encoding="utf-8" ?>
<wsdl:definitions xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:s1="http://microsoft.com/wsdl/types/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
targetNamespace="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">User Profile Service</wsdl:documentation>
  <wsdl:types>
    <s:schema elementFormDefault="qualified"
targetNamespace="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService">
      <s:import namespace="http://microsoft.com/wsdl/types/" />
      <s:element name="GetUserProfileByIndex">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="index" type="s:int" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetUserProfileByIndexResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1"
name="GetUserProfileByIndexResult" type="tns:GetUserProfileByIndexResult" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="GetUserProfileByIndexResult">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="NextValue" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="UserProfile"
type="tns:ArrayOfPropertyData" />
          <s:element minOccurs="0" maxOccurs="1" name="Colleagues"
type="tns:ArrayOfContactData" />
          <s:element minOccurs="0" maxOccurs="1" name="QuickLinks"
type="tns:ArrayOfQuickLinkData" />
          <s:element minOccurs="0" maxOccurs="1" name="PinnedLinks"
type="tns:ArrayOfPinnedLinkData" />
          <s:element minOccurs="0" maxOccurs="1" name="Memberships"
type="tns:ArrayOfMembershipData" />
        </s:sequence>
      </s:complexType>
      <s:complexType name="ArrayOfPropertyData">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="unbounded" name="PropertyData"
nillable="true" type="tns:PropertyData" />
        </s:sequence>
      </s:complexType>
    </s:schema>
  </wsdl:types>

```

```

    </s:complexType>
    <s:complexType name="PropertyData">
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="IsPrivacyChanged"
type="s:boolean" />
        <s:element minOccurs="1" maxOccurs="1" name="IsValueChanged"
type="s:boolean" />
        <s:element minOccurs="0" maxOccurs="1" name="Name" type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="Privacy" type="tns:Privacy"
/>
        <s:element minOccurs="0" maxOccurs="1" name="Values"
type="tns:ArrayOfValueData" />
      </s:sequence>
    </s:complexType>
    <s:simpleType name="Privacy">
      <s:restriction base="s:string">
        <s:enumeration value="Public" />
        <s:enumeration value="Contacts" />
        <s:enumeration value="Organization" />
        <s:enumeration value="Manager" />
        <s:enumeration value="Private" />
        <s:enumeration value="NotSet" />
      </s:restriction>
    </s:simpleType>
    <s:complexType name="ArrayOfValueData">
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="unbounded" name="ValueData"
nillable="true" type="tns:ValueData" />
      </s:sequence>
    </s:complexType>
    <s:complexType name="ValueData">
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="Value" type="s:string" />
      </s:sequence>
    </s:complexType>
    <s:complexType name="ArrayOfContactData">
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="unbounded" name="ContactData"
nillable="true" type="tns:ContactData" />
      </s:sequence>
    </s:complexType>
    <s:complexType name="ContactData">
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="AccountName" type="s:string"
/>
        <s:element minOccurs="1" maxOccurs="1" name="Privacy" type="tns:Privacy"
/>
        <s:element minOccurs="0" maxOccurs="1" name="Name" type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="IsInWorkGroup"
type="s:boolean" />
        <s:element minOccurs="0" maxOccurs="1" name="Group" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="Email" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="Title" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="Url" type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="UserProfileID"
type="s1:guid" />
        <s:element minOccurs="1" maxOccurs="1" name="ID" type="s:long" />
      </s:sequence>
    </s:complexType>
    <s:complexType name="ArrayOfQuickLinkData">
      <s:sequence>

```

```

        <s:element minOccurs="0" maxOccurs="unbounded" name="QuickLinkData"
nillable="true" type="tns:QuickLinkData" />
    </s:sequence>
</s:complexType>
<s:complexType name="QuickLinkData">
    <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="Name" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="Group" type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="Privacy" type="tns:Privacy"
/>
        <s:element minOccurs="0" maxOccurs="1" name="Url" type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="ID" type="s:long" />
    </s:sequence>
</s:complexType>
<s:complexType name="ArrayOfPinnedLinkData">
    <s:sequence>
        <s:element minOccurs="0" maxOccurs="unbounded" name="PinnedLinkData"
nillable="true" type="tns:PinnedLinkData" />
    </s:sequence>
</s:complexType>
<s:complexType name="PinnedLinkData">
    <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="Name" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="Url" type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="ID" type="s:long" />
    </s:sequence>
</s:complexType>
<s:complexType name="ArrayOfMembershipData">
    <s:sequence>
        <s:element minOccurs="0" maxOccurs="unbounded" name="MembershipData"
nillable="true" type="tns:MembershipData" />
    </s:sequence>
</s:complexType>
<s:complexType name="MembershipData">
    <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="Source"
type="tns:MembershipSource" />
        <s:element minOccurs="0" maxOccurs="1" name="MemberGroup"
type="tns:MemberGroupData" />
        <s:element minOccurs="0" maxOccurs="1" name="Group" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="DisplayName" type="s:string"
/>
        <s:element minOccurs="1" maxOccurs="1" name="Privacy" type="tns:Privacy"
/>
        <s:element minOccurs="0" maxOccurs="1" name="MailNickname"
type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="Url" type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="ID" type="s:long" />
        <s:element minOccurs="1" maxOccurs="1" name="MemberGroupID" type="s:long"
/>
    </s:sequence>
</s:complexType>
<s:simpleType name="MembershipSource">
    <s:restriction base="s:string">
        <s:enumeration value="DistributionList" />
        <s:enumeration value="SharePointSite" />
        <s:enumeration value="Other" />
    </s:restriction>
</s:simpleType>
<s:complexType name="MemberGroupData">
    <s:sequence>

```

```

        <s:element minOccurs="1" maxOccurs="1" name="SourceInternal"
type="s1:guid" />
        <s:element minOccurs="0" maxOccurs="1" name="SourceReference"
type="s:string" />
    </s:sequence>
</s:complexType>
<s:complexType name="SPTimeZone">
    <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="ID" type="s:unsignedShort"
/>
    </s:sequence>
</s:complexType>
<s:element name="CreateUserProfileByAccountName">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="accountName"
type="s:string" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="CreateUserProfileByAccountNameResponse">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1"
name="CreateUserProfileByAccountNameResult" type="tns:ArrayOfPropertyData" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="GetUserProfileByName">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="accountName"
type="s:string" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="GetUserProfileByNameResponse">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1"
name="GetUserProfileByNameResult" type="tns:ArrayOfPropertyData" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="GetUserProfileByGuid">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="guid" type="s1:guid" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="GetUserProfileByGuidResponse">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1"
name="GetUserProfileByGuidResult" type="tns:ArrayOfPropertyData" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="GetUserProfileSchema">
    <s:complexType />

```

```

    </s:element>
    <s:complexType name="PropertyInfo">
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="Name" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="Description" type="s:string" />
      />
      <s:element minOccurs="0" maxOccurs="1" name="DisplayOrder" type="s:int" />
      <s:element minOccurs="0" maxOccurs="1" name="MaximumShown" type="s:int" />
      <s:element minOccurs="0" maxOccurs="1" name="IsAdminEditable" type="s:boolean" />
      <s:element minOccurs="0" maxOccurs="1" name="IsSearchable" type="s:boolean" />
      <s:element minOccurs="0" maxOccurs="1" name="IsSystem" type="s:boolean" />
      <s:element minOccurs="0" maxOccurs="1" name="ManagedPropertyName" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="DisplayName" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="Type" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="AllowPolicyOverride" type="s:boolean" />
      <s:element minOccurs="1" maxOccurs="1" name="DefaultPrivacy" type="tns:Privacy" />
      <s:element minOccurs="1" maxOccurs="1" name="IsAlias" type="s:boolean" />
      <s:element minOccurs="1" maxOccurs="1" name="IsColleagueEventLog" type="s:boolean" />
      <s:element minOccurs="1" maxOccurs="1" name="IsRequired" type="s:boolean" />
      <s:element minOccurs="1" maxOccurs="1" name="IsUserEditable" type="s:boolean" />
      <s:element minOccurs="1" maxOccurs="1" name="IsVisibleOnEditor" type="s:boolean" />
      <s:element minOccurs="1" maxOccurs="1" name="IsVisibleOnViewer" type="s:boolean" />
      <s:element minOccurs="1" maxOccurs="1" name="IsReplicable" type="s:boolean" />
      <s:element minOccurs="1" maxOccurs="1" name="UserOverridePrivacy" type="s:boolean" />
      <s:element minOccurs="1" maxOccurs="1" name="Length" type="s:int" />
      <s:element minOccurs="1" maxOccurs="1" name="IsImported" type="s:boolean" />
      <s:element minOccurs="1" maxOccurs="1" name="IsMultiValue" type="s:boolean" />
      <s:element minOccurs="1" maxOccurs="1" name="ChoiceType" type="tns:ChoiceTypes" />
      <s:element minOccurs="0" maxOccurs="1" name="TermSetId" nillable="true" type="s1:guid" />
    </s:sequence>
  </s:complexType>
  <s:simpleType name="ChoiceTypes">
    <s:restriction base="s:string">
      <s:enumeration value="Off" />
      <s:enumeration value="None" />
      <s:enumeration value="Open" />
      <s:enumeration value="Closed" />
    </s:restriction>
  </s:simpleType>
  <s:complexType name="ArrayOfPropertyInfo">
    <s:sequence>

```

```

        <s:element minOccurs="0" maxOccurs="unbounded" name="PropertyInfo"
nillable="true" type="tns:PropertyInfo" />
    </s:sequence>
</s:complexType>
<s:element name="GetUserProfileSchemaResponse">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1"
name="GetUserProfileSchemaResult" type="tns:ArrayOfPropertyInfo" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="GetProfileSchemaNameByAccountName">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="accountName"
type="s:string" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="GetProfileSchemaNameByAccountNameResponse">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1"
name="GetProfileSchemaNameByAccountNameResult" type="s:string" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="GetPropertyChoiceList">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="propertyName"
type="s:string" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:complexType name="ArrayOfString">
    <s:sequence>
        <s:element minOccurs="0" maxOccurs="unbounded" name="string"
nillable="true" type="s:string" />
    </s:sequence>
</s:complexType>
<s:element name="GetPropertyChoiceListResponse">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1"
name="GetPropertyChoiceListResult" type="tns:ArrayOfString" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="ModifyUserPropertyByAccountName">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="accountName"
type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="newData"
type="tns:ArrayOfPropertyData" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="ModifyUserPropertyByAccountNameResponse">

```



```

        <s:complexType />
    </s:element>
    <s:element name="GetUserPropertyByAccountName">
        <s:complexType>
            <s:sequence>
                <s:element minOccurs="0" maxOccurs="1" name="accountName"
type="s:string" />
                <s:element minOccurs="0" maxOccurs="1" name="propertyName"
type="s:string" />
            </s:sequence>
        </s:complexType>
    </s:element>
    <s:element name="GetUserPropertyByAccountNameResponse">
        <s:complexType>
            <s:sequence>
                <s:element minOccurs="0" maxOccurs="1"
name="GetUserPropertyByAccountNameResult" type="tns:PropertyData" />
            </s:sequence>
        </s:complexType>
    </s:element>
    <s:element name="CreateMemberGroup">
        <s:complexType>
            <s:sequence>
                <s:element minOccurs="0" maxOccurs="1" name="membershipInfo"
type="tns:MembershipData" />
            </s:sequence>
        </s:complexType>
    </s:element>
    <s:element name="CreateMemberGroupResponse">
        <s:complexType />
    </s:element>
    <s:element name="AddMembership">
        <s:complexType>
            <s:sequence>
                <s:element minOccurs="0" maxOccurs="1" name="accountName"
type="s:string" />
                <s:element minOccurs="0" maxOccurs="1" name="membershipInfo"
type="tns:MembershipData" />
                <s:element minOccurs="0" maxOccurs="1" name="group" type="s:string" />
                <s:element minOccurs="1" maxOccurs="1" name="privacy"
type="tns:Privacy" />
            </s:sequence>
        </s:complexType>
    </s:element>
    <s:element name="AddMembershipResponse">
        <s:complexType>
            <s:sequence>
                <s:element minOccurs="0" maxOccurs="1" name="AddMembershipResult"
type="tns:MembershipData" />
            </s:sequence>
        </s:complexType>
    </s:element>
    <s:element name="RemoveMembership">
        <s:complexType>
            <s:sequence>
                <s:element minOccurs="0" maxOccurs="1" name="accountName"
type="s:string" />
                <s:element minOccurs="1" maxOccurs="1" name="sourceInternal"
type="s1:guid" />
                <s:element minOccurs="0" maxOccurs="1" name="sourceReference"
type="s:string" />
            </s:sequence>
        </s:complexType>
    </s:element>

```

```

        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="RemoveMembershipResponse">
    <s:complexType />
</s:element>
<s:element name="UpdateMembershipPrivacy">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="accountName"
type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="sourceInternal"
type="s1:guid" />
            <s:element minOccurs="0" maxOccurs="1" name="sourceReference"
type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="newPrivacy"
type="tns:Privacy" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="UpdateMembershipPrivacyResponse">
    <s:complexType />
</s:element>
<s:element name="GetUserMemberships">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="accountName"
type="s:string" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="GetUserMembershipsResponse">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="GetUserMembershipsResult"
type="tns:ArrayOfMembershipData" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="GetUserOrganizations">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="accountName"
type="s:string" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:complexType name="OrganizationProfileData">
    <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="DisplayName" type="s:string"
/>
        <s:element minOccurs="1" maxOccurs="1" name="RecordID" type="s:long" />
    </s:sequence>
</s:complexType>
<s:complexType name="ArrayOfOrganizationProfileData">
    <s:sequence>
        <s:element minOccurs="0" maxOccurs="unbounded"
name="OrganizationProfileData" nillable="true" type="tns:OrganizationProfileData"
/>
    </s:sequence>
</s:complexType>

```

```

    <s:element name="GetUserOrganizationsResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1"
name="GetUserOrganizationsResult" type="tns:ArrayOfOrganizationProfileData" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="GetUserColleagues">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="accountName"
type="s:string" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="GetUserColleaguesResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="GetUserColleaguesResult"
type="tns:ArrayOfContactData" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="GetUserLinks">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="accountName"
type="s:string" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="GetUserLinksResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="GetUserLinksResult"
type="tns:ArrayOfQuickLinkData" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="GetUserPinnedLinks">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="accountName"
type="s:string" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="GetUserPinnedLinksResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="GetUserPinnedLinksResult"
type="tns:ArrayOfPinnedLinkData" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="GetInCommon">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="accountName"
type="s:string" />

```

```

        </s:sequence>
    </s:complexType>
</s:element>
<s:complexType name="InCommonData">
    <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="Manager"
type="tns:ContactData" />
        <s:element minOccurs="0" maxOccurs="1" name="Colleagues"
type="tns:ArrayOfContactData" />
        <s:element minOccurs="0" maxOccurs="1" name="Memberships"
type="tns:ArrayOfMembershipData" />
    </s:sequence>
</s:complexType>
<s:element name="GetInCommonResponse">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="GetInCommonResult"
type="tns:InCommonData" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="GetCommonManager">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="accountName"
type="s:string" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="GetCommonManagerResponse">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="GetCommonManagerResult"
type="tns:ContactData" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="GetCommonColleagues">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="accountName"
type="s:string" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="GetCommonColleaguesResponse">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="GetCommonColleaguesResult"
type="tns:ArrayOfContactData" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="GetCommonMemberships">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="accountName"
type="s:string" />
        </s:sequence>
    </s:complexType>
</s:element>

```

```

<s:element name="GetCommonMembershipsResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1"
name="GetCommonMembershipsResult" type="tns:ArrayOfMembershipData" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="AddColleague">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="accountName"
type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="colleagueAccountName"
type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="group" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="privacy"
type="tns:Privacy" />
      <s:element minOccurs="1" maxOccurs="1" name="isInWorkGroup"
type="s:boolean" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="AddColleagueResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="AddColleagueResult"
type="tns:ContactData" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="AddColleagueWithoutEmailNotification">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="accountName"
type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="colleagueAccountName"
type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="group" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="privacy"
type="tns:Privacy" />
      <s:element minOccurs="1" maxOccurs="1" name="isInWorkGroup"
type="s:boolean" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="AddColleagueWithoutEmailNotificationResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1"
name="AddColleagueWithoutEmailNotificationResult" type="tns:ContactData" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="RemoveColleague">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="accountName"
type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="colleagueAccountName"
type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>

```

```

        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="RemoveColleagueResponse">
    <s:complexType />
</s:element>
<s:element name="UpdateColleaguePrivacy">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="accountName"
type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="colleagueAccountName"
type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="newPrivacy"
type="tns:Privacy" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="UpdateColleaguePrivacyResponse">
    <s:complexType />
</s:element>
<s:element name="AddPinnedLink">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="accountName"
type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="name" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="url" type="s:string" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="AddPinnedLinkResponse">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="AddPinnedLinkResult"
type="tns:PinnedLinkData" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="UpdatePinnedLink">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="accountName"
type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="data"
type="tns:PinnedLinkData" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="UpdatePinnedLinkResponse">
    <s:complexType />
</s:element>
<s:element name="RemovePinnedLink">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="accountName"
type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="id" type="s:int" />
        </s:sequence>
    </s:complexType>
</s:element>

```

```

    <s:element name="RemovePinnedLinkResponse">
      <s:complexType />
    </s:element>
    <s:element name="AddLink">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="accountName"
type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="name" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="url" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="group" type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="privacy"
type="tns:Privacy" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="AddLinkResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="AddLinkResult"
type="tns:QuickLinkData" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="UpdateLink">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="accountName"
type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="data"
type="tns:QuickLinkData" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="UpdateLinkResponse">
      <s:complexType />
    </s:element>
    <s:element name="RemoveLink">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="accountName"
type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="id" type="s:int" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="RemoveLinkResponse">
      <s:complexType />
    </s:element>
    <s:element name="RemoveAllLinks">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="accountName"
type="s:string" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="RemoveAllLinksResponse">
      <s:complexType />
    </s:element>
    <s:element name="RemoveAllPinnedLinks">

```

```

        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="accountName"
type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="RemoveAllPinnedLinksResponse">
        <s:complexType />
      </s:element>
      <s:element name="RemoveAllColleagues">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="accountName"
type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="RemoveAllColleaguesResponse">
        <s:complexType />
      </s:element>
      <s:element name="RemoveAllMemberships">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="accountName"
type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="RemoveAllMembershipsResponse">
        <s:complexType />
      </s:element>
      <s:element name="GetUserProfileCount">
        <s:complexType />
      </s:element>
      <s:element name="GetUserProfileCountResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="GetUserProfileCountResult"
type="s:long" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="AddSuggestions">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="type"
type="tns:SuggestionType" />
            <s:element minOccurs="0" maxOccurs="1" name="suggestions"
type="tns:ArrayOfString" />
            <s:element minOccurs="0" maxOccurs="1" name="weights"
type="tns:ArrayOfDouble" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:simpleType name="SuggestionType">
        <s:restriction base="s:string">
          <s:enumeration value="Colleague" />
          <s:enumeration value="Keyword" />
        </s:restriction>
      </s:simpleType>

```



```

    <s:complexType name="ArrayOfDouble">
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="unbounded" name="double"
type="s:double" />
      </s:sequence>
    </s:complexType>
    <s:element name="AddSuggestionsResponse">
      <s:complexType />
    </s:element>
    <s:element name="GetProfileSchemaNames">
      <s:complexType />
    </s:element>
    <s:element name="GetProfileSchemaNamesResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1"
name="GetProfileSchemaNamesResult" type="tns:ArrayOfString" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="GetProfileSchema">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="schemaName"
type="s:string" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="GetProfileSchemaResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="GetProfileSchemaResult"
type="tns:ArrayOfPropertyInfo" />
        </s:sequence>
      </s:complexType>
    </s:element>
  </s:schema>
  <s:schema elementFormDefault="qualified"
targetNamespace="http://microsoft.com/wsdl/types/">
    <s:simpleType name="guid">
      <s:restriction base="s:string">
        <s:pattern value="[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-
F]{4}-[0-9a-fA-F]{12}" />
      </s:restriction>
    </s:simpleType>
  </s:schema>
</wsdl:types>
<wsdl:message name="GetUserProfileByIndexSoapIn">
  <wsdl:part name="parameters" element="tns:GetUserProfileByIndex" />
</wsdl:message>
<wsdl:message name="GetUserProfileByIndexSoapOut">
  <wsdl:part name="parameters" element="tns:GetUserProfileByIndexResponse" />
</wsdl:message>
<wsdl:message name="CreateUserProfileByAccountNameSoapIn">
  <wsdl:part name="parameters" element="tns:CreateUserProfileByAccountName" />
</wsdl:message>
<wsdl:message name="CreateUserProfileByAccountNameSoapOut">
  <wsdl:part name="parameters"
element="tns:CreateUserProfileByAccountNameResponse" />
</wsdl:message>
<wsdl:message name="GetUserProfileByNameSoapIn">

```

```

    <wsdl:part name="parameters" element="tns:GetUserProfileByName" />
</wsdl:message>
<wsdl:message name="GetUserProfileByNameSoapOut">
    <wsdl:part name="parameters" element="tns:GetUserProfileByNameResponse" />
</wsdl:message>
<wsdl:message name="GetUserProfileByGuidSoapIn">
    <wsdl:part name="parameters" element="tns:GetUserProfileByGuid" />
</wsdl:message>
<wsdl:message name="GetUserProfileByGuidSoapOut">
    <wsdl:part name="parameters" element="tns:GetUserProfileByGuidResponse" />
</wsdl:message>
<wsdl:message name="GetUserProfileSchemaSoapIn">
    <wsdl:part name="parameters" element="tns:GetUserProfileSchema" />
</wsdl:message>
<wsdl:message name="GetUserProfileSchemaSoapOut">
    <wsdl:part name="parameters" element="tns:GetUserProfileSchemaResponse" />
</wsdl:message>
<wsdl:message name="GetProfileSchemaNameByAccountNameSoapIn">
    <wsdl:part name="parameters" element="tns:GetProfileSchemaNameByAccountName" />
</wsdl:message>
<wsdl:message name="GetProfileSchemaNameByAccountNameSoapOut">
    <wsdl:part name="parameters"
element="tns:GetProfileSchemaNameByAccountNameResponse" />
</wsdl:message>
<wsdl:message name="GetPropertyChoiceListSoapIn">
    <wsdl:part name="parameters" element="tns:GetPropertyChoiceList" />
</wsdl:message>
<wsdl:message name="GetPropertyChoiceListSoapOut">
    <wsdl:part name="parameters" element="tns:GetPropertyChoiceListResponse" />
</wsdl:message>
<wsdl:message name="ModifyUserPropertyByAccountNameSoapIn">
    <wsdl:part name="parameters" element="tns:ModifyUserPropertyByAccountName" />
</wsdl:message>
<wsdl:message name="ModifyUserPropertyByAccountNameSoapOut">
    <wsdl:part name="parameters"
element="tns:ModifyUserPropertyByAccountNameResponse" />
</wsdl:message>
<wsdl:message name="GetUserPropertyByAccountNameSoapIn">
    <wsdl:part name="parameters" element="tns:GetUserPropertyByAccountName" />
</wsdl:message>
<wsdl:message name="GetUserPropertyByAccountNameSoapOut">
    <wsdl:part name="parameters" element="tns:GetUserPropertyByAccountNameResponse"
/>
</wsdl:message>
<wsdl:message name="CreateMemberGroupSoapIn">
    <wsdl:part name="parameters" element="tns:CreateMemberGroup" />
</wsdl:message>
<wsdl:message name="CreateMemberGroupSoapOut">
    <wsdl:part name="parameters" element="tns:CreateMemberGroupResponse" />
</wsdl:message>
<wsdl:message name="AddMembershipSoapIn">
    <wsdl:part name="parameters" element="tns:AddMembership" />
</wsdl:message>
<wsdl:message name="AddMembershipSoapOut">
    <wsdl:part name="parameters" element="tns:AddMembershipResponse" />
</wsdl:message>
<wsdl:message name="RemoveMembershipSoapIn">
    <wsdl:part name="parameters" element="tns:RemoveMembership" />
</wsdl:message>
<wsdl:message name="RemoveMembershipSoapOut">
    <wsdl:part name="parameters" element="tns:RemoveMembershipResponse" />

```

```

</wsdl:message>
<wsdl:message name="UpdateMembershipPrivacySoapIn">
  <wsdl:part name="parameters" element="tns:UpdateMembershipPrivacy" />
</wsdl:message>
<wsdl:message name="UpdateMembershipPrivacySoapOut">
  <wsdl:part name="parameters" element="tns:UpdateMembershipPrivacyResponse" />
</wsdl:message>
<wsdl:message name="GetUserMembershipsSoapIn">
  <wsdl:part name="parameters" element="tns:GetUserMemberships" />
</wsdl:message>
<wsdl:message name="GetUserMembershipsSoapOut">
  <wsdl:part name="parameters" element="tns:GetUserMembershipsResponse" />
</wsdl:message>
<wsdl:message name="GetUserOrganizationsSoapIn">
  <wsdl:part name="parameters" element="tns:GetUserOrganizations" />
</wsdl:message>
<wsdl:message name="GetUserOrganizationsSoapOut">
  <wsdl:part name="parameters" element="tns:GetUserOrganizationsResponse" />
</wsdl:message>
<wsdl:message name="GetUserColleaguesSoapIn">
  <wsdl:part name="parameters" element="tns:GetUserColleagues" />
</wsdl:message>
<wsdl:message name="GetUserColleaguesSoapOut">
  <wsdl:part name="parameters" element="tns:GetUserColleaguesResponse" />
</wsdl:message>
<wsdl:message name="GetUserLinksSoapIn">
  <wsdl:part name="parameters" element="tns:GetUserLinks" />
</wsdl:message>
<wsdl:message name="GetUserLinksSoapOut">
  <wsdl:part name="parameters" element="tns:GetUserLinksResponse" />
</wsdl:message>
<wsdl:message name="GetUserPinnedLinksSoapIn">
  <wsdl:part name="parameters" element="tns:GetUserPinnedLinks" />
</wsdl:message>
<wsdl:message name="GetUserPinnedLinksSoapOut">
  <wsdl:part name="parameters" element="tns:GetUserPinnedLinksResponse" />
</wsdl:message>
<wsdl:message name="GetInCommonSoapIn">
  <wsdl:part name="parameters" element="tns:GetInCommon" />
</wsdl:message>
<wsdl:message name="GetInCommonSoapOut">
  <wsdl:part name="parameters" element="tns:GetInCommonResponse" />
</wsdl:message>
<wsdl:message name="GetCommonManagerSoapIn">
  <wsdl:part name="parameters" element="tns:GetCommonManager" />
</wsdl:message>
<wsdl:message name="GetCommonManagerSoapOut">
  <wsdl:part name="parameters" element="tns:GetCommonManagerResponse" />
</wsdl:message>
<wsdl:message name="GetCommonColleaguesSoapIn">
  <wsdl:part name="parameters" element="tns:GetCommonColleagues" />
</wsdl:message>
<wsdl:message name="GetCommonColleaguesSoapOut">
  <wsdl:part name="parameters" element="tns:GetCommonColleaguesResponse" />
</wsdl:message>
<wsdl:message name="GetCommonMembershipsSoapIn">
  <wsdl:part name="parameters" element="tns:GetCommonMemberships" />
</wsdl:message>
<wsdl:message name="GetCommonMembershipsSoapOut">
  <wsdl:part name="parameters" element="tns:GetCommonMembershipsResponse" />
</wsdl:message>

```

```

<wsdl:message name="AddColleagueSoapIn">
  <wsdl:part name="parameters" element="tns:AddColleague" />
</wsdl:message>
<wsdl:message name="AddColleagueSoapOut">
  <wsdl:part name="parameters" element="tns:AddColleagueResponse" />
</wsdl:message>
<wsdl:message name="AddColleagueWithoutEmailNotificationSoapIn">
  <wsdl:part name="parameters" element="tns:AddColleagueWithoutEmailNotification"
/>
</wsdl:message>
<wsdl:message name="AddColleagueWithoutEmailNotificationSoapOut">
  <wsdl:part name="parameters"
element="tns:AddColleagueWithoutEmailNotificationResponse" />
</wsdl:message>
<wsdl:message name="RemoveColleagueSoapIn">
  <wsdl:part name="parameters" element="tns:RemoveColleague" />
</wsdl:message>
<wsdl:message name="RemoveColleagueSoapOut">
  <wsdl:part name="parameters" element="tns:RemoveColleagueResponse" />
</wsdl:message>
<wsdl:message name="UpdateColleaguePrivacySoapIn">
  <wsdl:part name="parameters" element="tns:UpdateColleaguePrivacy" />
</wsdl:message>
<wsdl:message name="UpdateColleaguePrivacySoapOut">
  <wsdl:part name="parameters" element="tns:UpdateColleaguePrivacyResponse" />
</wsdl:message>
<wsdl:message name="AddPinnedLinkSoapIn">
  <wsdl:part name="parameters" element="tns:AddPinnedLink" />
</wsdl:message>
<wsdl:message name="AddPinnedLinkSoapOut">
  <wsdl:part name="parameters" element="tns:AddPinnedLinkResponse" />
</wsdl:message>
<wsdl:message name="UpdatePinnedLinkSoapIn">
  <wsdl:part name="parameters" element="tns:UpdatePinnedLink" />
</wsdl:message>
<wsdl:message name="UpdatePinnedLinkSoapOut">
  <wsdl:part name="parameters" element="tns:UpdatePinnedLinkResponse" />
</wsdl:message>
<wsdl:message name="RemovePinnedLinkSoapIn">
  <wsdl:part name="parameters" element="tns:RemovePinnedLink" />
</wsdl:message>
<wsdl:message name="RemovePinnedLinkSoapOut">
  <wsdl:part name="parameters" element="tns:RemovePinnedLinkResponse" />
</wsdl:message>
<wsdl:message name="AddLinkSoapIn">
  <wsdl:part name="parameters" element="tns:AddLink" />
</wsdl:message>
<wsdl:message name="AddLinkSoapOut">
  <wsdl:part name="parameters" element="tns:AddLinkResponse" />
</wsdl:message>
<wsdl:message name="UpdateLinkSoapIn">
  <wsdl:part name="parameters" element="tns:UpdateLink" />
</wsdl:message>
<wsdl:message name="UpdateLinkSoapOut">
  <wsdl:part name="parameters" element="tns:UpdateLinkResponse" />
</wsdl:message>
<wsdl:message name="RemoveLinkSoapIn">
  <wsdl:part name="parameters" element="tns:RemoveLink" />
</wsdl:message>
<wsdl:message name="RemoveLinkSoapOut">
  <wsdl:part name="parameters" element="tns:RemoveLinkResponse" />

```

```

</wsdl:message>
<wsdl:message name="RemoveAllLinksSoapIn">
  <wsdl:part name="parameters" element="tns:RemoveAllLinks" />
</wsdl:message>
<wsdl:message name="RemoveAllLinksSoapOut">
  <wsdl:part name="parameters" element="tns:RemoveAllLinksResponse" />
</wsdl:message>
<wsdl:message name="RemoveAllPinnedLinksSoapIn">
  <wsdl:part name="parameters" element="tns:RemoveAllPinnedLinks" />
</wsdl:message>
<wsdl:message name="RemoveAllPinnedLinksSoapOut">
  <wsdl:part name="parameters" element="tns:RemoveAllPinnedLinksResponse" />
</wsdl:message>
<wsdl:message name="RemoveAllColleaguesSoapIn">
  <wsdl:part name="parameters" element="tns:RemoveAllColleagues" />
</wsdl:message>
<wsdl:message name="RemoveAllColleaguesSoapOut">
  <wsdl:part name="parameters" element="tns:RemoveAllColleaguesResponse" />
</wsdl:message>
<wsdl:message name="RemoveAllMembershipsSoapIn">
  <wsdl:part name="parameters" element="tns:RemoveAllMemberships" />
</wsdl:message>
<wsdl:message name="RemoveAllMembershipsSoapOut">
  <wsdl:part name="parameters" element="tns:RemoveAllMembershipsResponse" />
</wsdl:message>
<wsdl:message name="GetUserProfileCountSoapIn">
  <wsdl:part name="parameters" element="tns:GetUserProfileCount" />
</wsdl:message>
<wsdl:message name="GetUserProfileCountSoapOut">
  <wsdl:part name="parameters" element="tns:GetUserProfileCountResponse" />
</wsdl:message>
<wsdl:message name="AddSuggestionsSoapIn">
  <wsdl:part name="parameters" element="tns:AddSuggestions" />
</wsdl:message>
<wsdl:message name="AddSuggestionsSoapOut">
  <wsdl:part name="parameters" element="tns:AddSuggestionsResponse" />
</wsdl:message>
<wsdl:message name="GetProfileSchemaNamesSoapIn">
  <wsdl:part name="parameters" element="tns:GetProfileSchemaNames" />
</wsdl:message>
<wsdl:message name="GetProfileSchemaNamesSoapOut">
  <wsdl:part name="parameters" element="tns:GetProfileSchemaNamesResponse" />
</wsdl:message>
<wsdl:message name="GetProfileSchemaSoapIn">
  <wsdl:part name="parameters" element="tns:GetProfileSchema" />
</wsdl:message>
<wsdl:message name="GetProfileSchemaSoapOut">
  <wsdl:part name="parameters" element="tns:GetProfileSchemaResponse" />
</wsdl:message>
<wsdl:portType name="UserProfileServiceSoap">
  <wsdl:operation name="GetUserProfileByIndex">
    <wsdl:input message="tns:GetUserProfileByIndexSoapIn" />
    <wsdl:output message="tns:GetUserProfileByIndexSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="CreateUserProfileByAccountName">
    <wsdl:input message="tns:CreateUserProfileByAccountNameSoapIn" />
    <wsdl:output message="tns:CreateUserProfileByAccountNameSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="GetUserProfileByName">
    <wsdl:input message="tns:GetUserProfileByNameSoapIn" />
    <wsdl:output message="tns:GetUserProfileByNameSoapOut" />
  </wsdl:operation>

```

```

</wsdl:operation>
<wsdl:operation name="GetUserProfileByGuid">
  <wsdl:input message="tns:GetUserProfileByGuidSoapIn" />
  <wsdl:output message="tns:GetUserProfileByGuidSoapOut" />
</wsdl:operation>
<wsdl:operation name="GetUserProfileSchema">
  <wsdl:input message="tns:GetUserProfileSchemaSoapIn" />
  <wsdl:output message="tns:GetUserProfileSchemaSoapOut" />
</wsdl:operation>
<wsdl:operation name="GetProfileSchemaNameByAccountName">
  <wsdl:input message="tns:GetProfileSchemaNameByAccountNameSoapIn" />
  <wsdl:output message="tns:GetProfileSchemaNameByAccountNameSoapOut" />
</wsdl:operation>
<wsdl:operation name="GetPropertyChoiceList">
  <wsdl:input message="tns:GetPropertyChoiceListSoapIn" />
  <wsdl:output message="tns:GetPropertyChoiceListSoapOut" />
</wsdl:operation>
<wsdl:operation name="ModifyUserPropertyByAccountName">
  <wsdl:input message="tns:ModifyUserPropertyByAccountNameSoapIn" />
  <wsdl:output message="tns:ModifyUserPropertyByAccountNameSoapOut" />
</wsdl:operation>
<wsdl:operation name="GetUserPropertyByAccountName">
  <wsdl:input message="tns:GetUserPropertyByAccountNameSoapIn" />
  <wsdl:output message="tns:GetUserPropertyByAccountNameSoapOut" />
</wsdl:operation>
<wsdl:operation name="CreateMemberGroup">
  <wsdl:input message="tns:CreateMemberGroupSoapIn" />
  <wsdl:output message="tns:CreateMemberGroupSoapOut" />
</wsdl:operation>
<wsdl:operation name="AddMembership">
  <wsdl:input message="tns:AddMembershipSoapIn" />
  <wsdl:output message="tns:AddMembershipSoapOut" />
</wsdl:operation>
<wsdl:operation name="RemoveMembership">
  <wsdl:input message="tns:RemoveMembershipSoapIn" />
  <wsdl:output message="tns:RemoveMembershipSoapOut" />
</wsdl:operation>
<wsdl:operation name="UpdateMembershipPrivacy">
  <wsdl:input message="tns:UpdateMembershipPrivacySoapIn" />
  <wsdl:output message="tns:UpdateMembershipPrivacySoapOut" />
</wsdl:operation>
<wsdl:operation name="GetUserMemberships">
  <wsdl:input message="tns:GetUserMembershipsSoapIn" />
  <wsdl:output message="tns:GetUserMembershipsSoapOut" />
</wsdl:operation>
<wsdl:operation name="GetUserOrganizations">
  <wsdl:input message="tns:GetUserOrganizationsSoapIn" />
  <wsdl:output message="tns:GetUserOrganizationsSoapOut" />
</wsdl:operation>
<wsdl:operation name="GetUserColleagues">
  <wsdl:input message="tns:GetUserColleaguesSoapIn" />
  <wsdl:output message="tns:GetUserColleaguesSoapOut" />
</wsdl:operation>
<wsdl:operation name="GetUserLinks">
  <wsdl:input message="tns:GetUserLinksSoapIn" />
  <wsdl:output message="tns:GetUserLinksSoapOut" />
</wsdl:operation>
<wsdl:operation name="GetUserPinnedLinks">
  <wsdl:input message="tns:GetUserPinnedLinksSoapIn" />
  <wsdl:output message="tns:GetUserPinnedLinksSoapOut" />
</wsdl:operation>

```

```

<wsdl:operation name="GetInCommon">
  <wsdl:input message="tns:GetInCommonSoapIn" />
  <wsdl:output message="tns:GetInCommonSoapOut" />
</wsdl:operation>
<wsdl:operation name="GetCommonManager">
  <wsdl:input message="tns:GetCommonManagerSoapIn" />
  <wsdl:output message="tns:GetCommonManagerSoapOut" />
</wsdl:operation>
<wsdl:operation name="GetCommonColleagues">
  <wsdl:input message="tns:GetCommonColleaguesSoapIn" />
  <wsdl:output message="tns:GetCommonColleaguesSoapOut" />
</wsdl:operation>
<wsdl:operation name="GetCommonMemberships">
  <wsdl:input message="tns:GetCommonMembershipsSoapIn" />
  <wsdl:output message="tns:GetCommonMembershipsSoapOut" />
</wsdl:operation>
<wsdl:operation name="AddColleague">
  <wsdl:input message="tns:AddColleagueSoapIn" />
  <wsdl:output message="tns:AddColleagueSoapOut" />
</wsdl:operation>
<wsdl:operation name="AddColleagueWithoutEmailNotification">
  <wsdl:input message="tns:AddColleagueWithoutEmailNotificationSoapIn" />
  <wsdl:output message="tns:AddColleagueWithoutEmailNotificationSoapOut" />
</wsdl:operation>
<wsdl:operation name="RemoveColleague">
  <wsdl:input message="tns:RemoveColleagueSoapIn" />
  <wsdl:output message="tns:RemoveColleagueSoapOut" />
</wsdl:operation>
<wsdl:operation name="UpdateColleaguePrivacy">
  <wsdl:input message="tns:UpdateColleaguePrivacySoapIn" />
  <wsdl:output message="tns:UpdateColleaguePrivacySoapOut" />
</wsdl:operation>
<wsdl:operation name="AddPinnedLink">
  <wsdl:input message="tns:AddPinnedLinkSoapIn" />
  <wsdl:output message="tns:AddPinnedLinkSoapOut" />
</wsdl:operation>
<wsdl:operation name="UpdatePinnedLink">
  <wsdl:input message="tns:UpdatePinnedLinkSoapIn" />
  <wsdl:output message="tns:UpdatePinnedLinkSoapOut" />
</wsdl:operation>
<wsdl:operation name="RemovePinnedLink">
  <wsdl:input message="tns:RemovePinnedLinkSoapIn" />
  <wsdl:output message="tns:RemovePinnedLinkSoapOut" />
</wsdl:operation>
<wsdl:operation name="AddLink">
  <wsdl:input message="tns:AddLinkSoapIn" />
  <wsdl:output message="tns:AddLinkSoapOut" />
</wsdl:operation>
<wsdl:operation name="UpdateLink">
  <wsdl:input message="tns:UpdateLinkSoapIn" />
  <wsdl:output message="tns:UpdateLinkSoapOut" />
</wsdl:operation>
<wsdl:operation name="RemoveLink">
  <wsdl:input message="tns:RemoveLinkSoapIn" />
  <wsdl:output message="tns:RemoveLinkSoapOut" />
</wsdl:operation>
<wsdl:operation name="RemoveAllLinks">
  <wsdl:input message="tns:RemoveAllLinksSoapIn" />
  <wsdl:output message="tns:RemoveAllLinksSoapOut" />
</wsdl:operation>
<wsdl:operation name="RemoveAllPinnedLinks">

```

```

        <wsdl:input message="tns:RemoveAllPinnedLinksSoapIn" />
        <wsdl:output message="tns:RemoveAllPinnedLinksSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="RemoveAllColleagues">
        <wsdl:input message="tns:RemoveAllColleaguesSoapIn" />
        <wsdl:output message="tns:RemoveAllColleaguesSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="RemoveAllMemberships">
        <wsdl:input message="tns:RemoveAllMembershipsSoapIn" />
        <wsdl:output message="tns:RemoveAllMembershipsSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="GetUserProfileCount">
        <wsdl:input message="tns:GetUserProfileCountSoapIn" />
        <wsdl:output message="tns:GetUserProfileCountSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="AddSuggestions">
        <wsdl:input message="tns:AddSuggestionsSoapIn" />
        <wsdl:output message="tns:AddSuggestionsSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="GetProfileSchemaNames">
        <wsdl:input message="tns:GetProfileSchemaNamesSoapIn" />
        <wsdl:output message="tns:GetProfileSchemaNamesSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="GetProfileSchema">
        <wsdl:input message="tns:GetProfileSchemaSoapIn" />
        <wsdl:output message="tns:GetProfileSchemaSoapOut" />
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="UserProfileServiceSoap" type="tns:UserProfileServiceSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="GetUserProfileByIndex">
        <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetUserProfileByIndex" style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="CreateUserProfileByAccountName">
        <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/CreateUserProfileByAccountName" style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetUserProfileByName">
        <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetUserProfileByName" style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>

```



```

    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetUserProfileByGuid">
    <soap:operation
      soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetUserProfileByGuid" style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetUserProfileSchema">
    <soap:operation
      soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetUserProfileSchema" style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetProfileSchemaNameByAccountName">
    <soap:operation
      soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetProfileSchemaNameByAccountName" style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetPropertyChoiceList">
    <soap:operation
      soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetPropertyChoiceList" style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="ModifyUserPropertyByAccountName">
    <soap:operation
      soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/ModifyUserPropertyByAccountName" style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetUserPropertyByAccountName">
    <soap:operation
      soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetUserPropertyByAccountName" style="document" />

```

```

    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" />
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="CreateMemberGroup">
  <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/CreateMemberGroup" style="document" />
  <wsdl:input>
    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" />
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="AddMembership">
  <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/AddMembership" style="document" />
  <wsdl:input>
    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" />
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="RemoveMembership">
  <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/RemoveMembership" style="document" />
  <wsdl:input>
    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" />
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="UpdateMembershipPrivacy">
  <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/UpdateMembershipPrivacy" style="document" />
  <wsdl:input>
    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" />
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetUserMemberships">
  <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetUserMemberships" style="document" />
  <wsdl:input>
    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" />
  </wsdl:output>

```

```

    </wsdl:operation>
    <wsdl:operation name="GetUserOrganizations">
      <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetUserOrganizations" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetUserColleagues">
      <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetUserColleagues" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetUserLinks">
      <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetUserLinks" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetUserPinnedLinks">
      <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetUserPinnedLinks" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetInCommon">
      <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetInCommon" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetCommonManager">
      <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetCommonManager" style="document" />
      <wsdl:input>

```

```

        <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetCommonColleagues">
    <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetCommonColleagues" style="document" />
    <wsdl:input>
        <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetCommonMemberships">
    <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetCommonMemberships" style="document" />
    <wsdl:input>
        <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="AddColleague">
    <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/AddColleague" style="document" />
    <wsdl:input>
        <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="AddColleagueWithoutEmailNotification">
    <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/AddColleagueWithoutEmailNotification" style="document" />
    <wsdl:input>
        <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="RemoveColleague">
    <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/RemoveColleague" style="document" />
    <wsdl:input>
        <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal" />
    </wsdl:output>
</wsdl:operation>

```

```

    <wsdl:operation name="UpdateColleaguePrivacy">
      <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/UpdateColleaguePrivacy" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="AddPinnedLink">
      <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/AddPinnedLink" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="UpdatePinnedLink">
      <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/UpdatePinnedLink" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="RemovePinnedLink">
      <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/RemovePinnedLink" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="AddLink">
      <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/AddLink" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="UpdateLink">
      <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/UpdateLink" style="document" />
      <wsdl:input>
        <soap:body use="literal" />

```

```

        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="RemoveLink">
        <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/RemoveLink" style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="RemoveAllLinks">
        <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/RemoveAllLinks" style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="RemoveAllPinnedLinks">
        <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/RemoveAllPinnedLinks" style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="RemoveAllColleagues">
        <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/RemoveAllColleagues" style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="RemoveAllMemberships">
        <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/RemoveAllMemberships" style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetUserProfileCount">

```

```

    <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetUserProfileCount" style="document" />
    <wsdl:input>
        <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="AddSuggestions">
    <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/AddSuggestions" style="document" />
    <wsdl:input>
        <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetProfileSchemaNames">
    <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetProfileSchemaNames" style="document" />
    <wsdl:input>
        <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetProfileSchema">
    <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetProfileSchema" style="document" />
    <wsdl:input>
        <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal" />
    </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:binding name="UserProfileServiceSoap12" type="tns:UserProfileServiceSoap">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="GetUserProfileByIndex">
        <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetUserProfileByIndex" style="document" />
        <wsdl:input>
            <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="CreateUserProfileByAccountName">
        <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/CreateUserProfileByAccountName" style="document" />

```

```

    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
  </wsdl:output>
    <soap12:body use="literal" />
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetUserProfileByName">
  <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetUserProfileByName" style="document" />
  <wsdl:input>
    <soap12:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal" />
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetUserProfileByGuid">
  <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetUserProfileByGuid" style="document" />
  <wsdl:input>
    <soap12:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal" />
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetUserProfileSchema">
  <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetUserProfileSchema" style="document" />
  <wsdl:input>
    <soap12:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal" />
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetProfileSchemaNameByAccountName">
  <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetProfileSchemaNameByAccountName" style="document" />
  <wsdl:input>
    <soap12:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal" />
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetPropertyChoiceList">
  <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetPropertyChoiceList" style="document" />
  <wsdl:input>
    <soap12:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal" />
  </wsdl:output>

```



```

    </wsdl:operation>
    <wsdl:operation name="ModifyUserPropertyByAccountName">
      <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/ModifyUserPropertyByAccountName" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetUserPropertyByAccountName">
      <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetUserPropertyByAccountName" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="CreateMemberGroup">
      <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/CreateMemberGroup" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="AddMembership">
      <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/AddMembership" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="RemoveMembership">
      <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/RemoveMembership" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="UpdateMembershipPrivacy">
      <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/UpdateMembershipPrivacy" style="document" />
      <wsdl:input>

```

```

        <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetUserMemberships">
    <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetUserMemberships" style="document" />
        <wsdl:input>
            <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
<wsdl:operation name="GetUserOrganizations">
    <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetUserOrganizations" style="document" />
        <wsdl:input>
            <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
<wsdl:operation name="GetUserColleagues">
    <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetUserColleagues" style="document" />
        <wsdl:input>
            <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
<wsdl:operation name="GetUserLinks">
    <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetUserLinks" style="document" />
        <wsdl:input>
            <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
<wsdl:operation name="GetUserPinnedLinks">
    <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetUserPinnedLinks" style="document" />
        <wsdl:input>
            <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" />
        </wsdl:output>
    </wsdl:operation>

```

```

    <wsdl:operation name="GetInCommon">
      <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileServ
ice/GetInCommon" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetCommonManager">
      <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileServ
ice/GetCommonManager" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetCommonColleagues">
      <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileServ
ice/GetCommonColleagues" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetCommonMemberships">
      <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileServ
ice/GetCommonMemberships" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="AddColleague">
      <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileServ
ice/AddColleague" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="AddColleagueWithoutEmailNotification">
      <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileServ
ice/AddColleagueWithoutEmailNotification" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />

```

```

        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="RemoveColleague">
        <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/RemoveColleague" style="document" />
        <wsdl:input>
            <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="UpdateColleaguePrivacy">
        <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/UpdateColleaguePrivacy" style="document" />
        <wsdl:input>
            <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="AddPinnedLink">
        <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/AddPinnedLink" style="document" />
        <wsdl:input>
            <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="UpdatePinnedLink">
        <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/UpdatePinnedLink" style="document" />
        <wsdl:input>
            <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="RemovePinnedLink">
        <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/RemovePinnedLink" style="document" />
        <wsdl:input>
            <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="AddLink">

```

```

    <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/AddLink" style="document" />
    <wsdl:input>
        <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="UpdateLink">
    <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/UpdateLink" style="document" />
    <wsdl:input>
        <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="RemoveLink">
    <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/RemoveLink" style="document" />
    <wsdl:input>
        <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="RemoveAllLinks">
    <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/RemoveAllLinks" style="document" />
    <wsdl:input>
        <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="RemoveAllPinnedLinks">
    <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/RemoveAllPinnedLinks" style="document" />
    <wsdl:input>
        <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="RemoveAllColleagues">
    <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/RemoveAllColleagues" style="document" />
    <wsdl:input>
        <soap12:body use="literal" />
    </wsdl:input>

```

```

        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="RemoveAllMemberships">
        <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/RemoveAllMemberships" style="document" />
        <wsdl:input>
          <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="GetUserProfileCount">
        <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetUserProfileCount" style="document" />
        <wsdl:input>
          <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="AddSuggestions">
        <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/AddSuggestions" style="document" />
        <wsdl:input>
          <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="GetProfileSchemaNames">
        <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetProfileSchemaNames" style="document" />
        <wsdl:input>
          <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="GetProfileSchema">
        <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/UserProfileService/GetProfileSchema" style="document" />
        <wsdl:input>
          <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
    </wsdl:binding>

```

</wsdl:definitions>

## 7 Appendix B: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include updates to those products.

- Microsoft Office Communicator 2007
- Microsoft Lync 2010
- Microsoft Lync Client 2013/Skype for Business
- The 2007 Microsoft Office system
- Microsoft Office 2010 suites
- Microsoft Office 2013
- Microsoft Office SharePoint Designer 2007
- Microsoft SharePoint Designer 2013
- Microsoft Office SharePoint Server 2007
- Microsoft SharePoint Server 2010
- Microsoft SharePoint Server 2013
- Microsoft SharePoint Workspace 2010
- Microsoft Office Visio 2007
- Microsoft Visio 2010
- Microsoft Visio 2013
- Microsoft Office 2016
- Microsoft Visio 2016
- Microsoft SharePoint Server 2016
- Microsoft Office 2019
- Microsoft SharePoint Server 2019
- Microsoft Visio 2019
- Microsoft Office 2021
- Microsoft Visio 2021
- Microsoft SharePoint Server Subscription Edition Preview
- Microsoft Skype for Business 2021

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates 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.

[<1> Section 3.1.4](#): Section 3.1.4: AddSuggestions is not supported in Office SharePoint Server 2007.

[<2> Section 3.1.4](#): Section 3.1.4: GetProfileSchema is not supported in Office SharePoint Server 2007.

[<3> Section 3.1.4](#): Section 3.1.4: GetProfileSchemaNameByAccountName is not supported in Office SharePoint Server 2007.

[<4> Section 3.1.4](#): Section 3.1.4: GetProfileSchemaNames is not supported in Office SharePoint Server 2007.

[<5> Section 3.1.4](#): Section 3.1.4: GetUserOrganizations is not supported in Office SharePoint Server 2007.

[<6> Section 3.1.4](#): Section 3.1.4: GetUserPropertyByAccountName is not supported in Office SharePoint Server 2007.

[<7> Section 3.1.4.4.2.1](#): SharePoint Server 2010 does not create a new user profile when **accountName** is null or the empty string.

[<8> Section 3.1.4.11.2.1](#): SharePoint Server 2010 does not create a new user profile when **accountName** is null or the empty string.

[<9> Section 3.1.4.12.2.2](#): Office SharePoint Server 2007 sends a SOAP fault in this case, SharePoint Server 2010 returns a ContactData with Manager set to null.

[<10> Section 3.1.4.16.2.1](#): In SharePoint Server 2010 and Office SharePoint Server 2007, a property can be unavailable through the web service for reasons such as, but not limited to: property is binary, property is managed by AD, property is a section. The reason is not specified by the web service.

[<11> Section 3.1.4.18.2.1](#): SharePoint Server 2010 does not create a new user profile when **accountName** is the empty string.

[<12> Section 3.1.4.19.2.1](#): SharePoint Server 2010 does not create a new user profile when **accountName** is the empty string.

[<13> Section 3.1.4.28.2.1](#): SharePoint Server 2010 does not create a new user profile when **accountName** is null.

[<14> Section 3.1.4.32.2.1](#): SharePoint Server 2010 does not create a new user profile when **accountName** is null or the empty string.

[<15> Section 3.1.4.36.2.1](#): SharePoint Server 2010 does not create a new user profile when **accountName** is null or the empty string.

[<16> Section 3.1.4.37.2.1](#): SharePoint Server 2010 does not create a new user profile when **accountName** is null or the empty string.

[<17> Section 3.1.4.38.2.1](#): SharePoint Server 2010 does not create a new user profile when **accountName** is null or the empty string.

## 8 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as Major, Minor, or None.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements.
- A document revision that captures changes to protocol functionality.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **None** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the relevant technical content is identical to the last released version.

The changes made to this document are listed in the following table. For more information, please contact [dochelp@microsoft.com](mailto:dochelp@microsoft.com).

Section	Description	Revision class
<a href="#">Z</a> Appendix B: Product Behavior	Updated list of supported products.	major
7 Appendix B: Product Behavior	Updated list of supported products.	Major

## 9 Index

### A

Abstract data model  
  [server](#) 29  
[Add memberships example](#) 119  
[Applicability](#) 15  
[ArrayOfContactData complex type](#) 17  
[ArrayOfDouble complex type](#) 17  
[ArrayOfMembershipData complex type](#) 18  
[ArrayOfOrganizationProfileData complex type](#) 18  
[ArrayOfPinnedLinkData complex type](#) 18  
[ArrayOfPropertyData complex type](#) 18  
[ArrayOfPropertyInfo complex type](#) 19  
[ArrayOfQuickLinkData complex type](#) 19  
[ArrayOfString complex type](#) 19  
[ArrayOfValueData complex type](#) 19  
[Attribute groups](#) 28  
[Attributes](#) 28

### C

[Capability negotiation](#) 15  
[Change tracking](#) 162  
[ChoiceTypes simple type](#) 25  
Client  
  [overview](#) 29  
[Commonalities example](#) 105  
[Complex types](#) 16  
  [ArrayOfContactData](#) 17  
  [ArrayOfDouble](#) 17  
  [ArrayOfMembershipData](#) 18  
  [ArrayOfOrganizationProfileData](#) 18  
  [ArrayOfPinnedLinkData](#) 18  
  [ArrayOfPropertyData](#) 18  
  [ArrayOfPropertyInfo](#) 19  
  [ArrayOfQuickLinkData](#) 19  
  [ArrayOfString](#) 19  
  [ArrayOfValueData](#) 19  
  [ContactData](#) 20  
  [MemberGroupData](#) 20  
  [MembershipData](#) 21  
  [OrganizationProfileData](#) 22  
  [PinnedLinkData](#) 22  
  [PropertyData](#) 22  
  [PropertyInfo](#) 23  
  [QuickLinkData](#) 24  
  [ValueData](#) 25  
[ContactData complex type](#) 20  
[Create the profile example](#) 104  
[Create user profile example](#) 104

### D

Data model - abstract  
  [server](#) 29  
[Delete memberships example](#) 120

### E

[Enumerate users example](#) 112  
Events

[local - server](#) 103  
[timer - server](#) 103

### Examples

[add memberships](#) 119  
[commonalities](#) 105  
[create the profile](#) 104  
[create user profile](#) 104  
[delete memberships](#) 120  
[enumerate users](#) 112  
[GetCommonColleagues](#) 107  
[GetCommonManager](#) 105  
[GetCommonMemberships](#) 106  
[GetInCommon](#) 108  
[GetUserProfile](#) 109  
[GetUserProfileByIndex](#) 112  
memberships ([section 4.5](#) 116, [section 4.5.1](#) 116)  
[obtain memberships](#) 117  
[obtain the first user](#) 112  
[obtain the profile of the specified user](#) 111  
[obtain the schema](#) 110  
[retrieve profile definition](#) 110  
[synchronize two users](#) 117

### F

[Fields - vendor-extensible](#) 15  
[Full WSDL](#) 123

### G

[GetCommonColleagues example](#) 107  
[GetCommonManager example](#) 105  
[GetCommonMemberships example](#) 106  
[GetInCommon example](#) 108  
[GetUserProfile example](#) 109  
[GetUserProfileByIndex example](#) 112  
[Glossary](#) 11  
[Groups](#) 28  
[guid simple type](#) 26

### I

[Implementer - security considerations](#) 122  
[Index of security parameters](#) 122  
[Informative references](#) 14  
Initialization  
  [server](#) 32  
[Introduction](#) 11

### L

Local events  
  [server](#) 103

### M

[MemberGroupData complex type](#) 20  
[MembershipData complex type](#) 21  
Memberships example ([section 4.5](#) 116, [section 4.5.1](#) 116)  
[MembershipSource simple type](#) 26

Message processing	
<a href="#">server</a>	33
Messages	
<a href="#">ArrayOfContactData complex type</a>	17
<a href="#">ArrayOfDouble complex type</a>	17
<a href="#">ArrayOfMembershipData complex type</a>	18
<a href="#">ArrayOfOrganizationProfileData complex type</a>	18
<a href="#">ArrayOfPinnedLinkData complex type</a>	18
<a href="#">ArrayOfPropertyData complex type</a>	18
<a href="#">ArrayOfPropertyInfo complex type</a>	19
<a href="#">ArrayOfQuickLinkData complex type</a>	19
<a href="#">ArrayOfString complex type</a>	19
<a href="#">ArrayOfValueData complex type</a>	19
<a href="#">attribute groups</a>	28
<a href="#">attributes</a>	28
<a href="#">ChoiceTypes simple type</a>	25
<a href="#">complex types</a>	16
<a href="#">ContactData complex type</a>	20
<a href="#">elements</a>	16
<a href="#">enumerated</a>	16
<a href="#">groups</a>	28
<a href="#">guid simple type</a>	26
<a href="#">MemberGroupData complex type</a>	20
<a href="#">MembershipData complex type</a>	21
<a href="#">MembershipSource simple type</a>	26
<a href="#">namespaces</a>	16
<a href="#">OrganizationProfileData complex type</a>	22
<a href="#">PinnedLinkData complex type</a>	22
<a href="#">Privacy simple type</a>	27
<a href="#">PropertyData complex type</a>	22
<a href="#">PropertyInfo complex type</a>	23
<a href="#">QuickLinkData complex type</a>	24
<a href="#">simple types</a>	25
<a href="#">SuggestionType simple type</a>	27
<a href="#">syntax</a>	16
<a href="#">transport</a>	16
<a href="#">ValueData complex type</a>	25
<b>N</b>	
<a href="#">Namespaces</a>	16
<a href="#">Normative references</a>	13
<b>O</b>	
<a href="#">Obtain memberships example</a>	117
<a href="#">Obtain the first user example</a>	112
<a href="#">Obtain the profile of the specified user example</a>	111
<a href="#">Obtain the schema example</a>	110
Operations	
<a href="#">AddColleague</a>	34
<a href="#">AddColleagueWithoutEmailNotification</a>	37
<a href="#">AddLink</a>	39
<a href="#">AddMembership</a>	41
<a href="#">AddPinnedLink</a>	42
<a href="#">AddSuggestions</a>	44
<a href="#">CreateMemberGroup</a>	46
<a href="#">CreateUserProfileByAccountName</a>	47
<a href="#">GetCommonColleagues</a>	49
<a href="#">GetCommonManager</a>	50
<a href="#">GetCommonMemberships</a>	52
<a href="#">GetInCommon</a>	54
<a href="#">GetProfileSchema</a>	56
<a href="#">GetProfileSchemaNameByAccountName</a>	57
<a href="#">GetProfileSchemaNames</a>	59
<a href="#">GetPropertyChoiceList</a>	60
<a href="#">GetUserColleagues</a>	62
<a href="#">GetUserLinks</a>	64
<a href="#">GetUserMemberships</a>	65
<a href="#">GetUserOrganizations</a>	67
<a href="#">GetUserPinnedLinks</a>	69
<a href="#">GetUserProfileByGuid</a>	71
<a href="#">GetUserProfileByIndex</a>	72
<a href="#">GetUserProfileByName</a>	75
<a href="#">GetUserProfileCount</a>	77
<a href="#">GetUserProfileSchema</a>	78
<a href="#">GetUserPropertyByAccountName</a>	79
<a href="#">ModifyUserPropertyByAccountName</a>	81
<a href="#">RemoveAllColleagues</a>	83
<a href="#">RemoveAllLinks</a>	84
<a href="#">RemoveAllMemberships</a>	86
<a href="#">RemoveAllPinnedLinks</a>	87
<a href="#">RemoveColleague</a>	89
<a href="#">RemoveLink</a>	90
<a href="#">RemoveMembership</a>	92
<a href="#">RemovePinnedLink</a>	94
<a href="#">UpdateColleaguePrivacy</a>	95
<a href="#">UpdateLink</a>	97
<a href="#">UpdateMembershipPrivacy</a>	99
<a href="#">UpdatePinnedLink</a>	101
<a href="#">OrganizationProfileData complex type</a>	22
<a href="#">Overview (synopsis)</a>	14
<b>P</b>	
<a href="#">Parameters - security index</a>	122
<a href="#">PinnedLinkData complex type</a>	22
<a href="#">Preconditions</a>	15
<a href="#">Prerequisites</a>	15
<a href="#">Privacy simple type</a>	27
<a href="#">Product behavior</a>	160
<a href="#">PropertyData complex type</a>	22
<a href="#">PropertyInfo complex type</a>	23
Protocol Details	
<a href="#">overview</a>	29
<b>Q</b>	
<a href="#">QuickLinkData complex type</a>	24
<b>R</b>	
<a href="#">References</a>	13
<a href="#">informative</a>	14
<a href="#">normative</a>	13
<a href="#">Relationship to other protocols</a>	14
<a href="#">Retrieve profile definition example</a>	110
<b>S</b>	
Security	
<a href="#">implementer considerations</a>	122
<a href="#">parameter index</a>	122
Sequencing rules	
<a href="#">server</a>	33
Server	
<a href="#">abstract data model</a>	29
<a href="#">AddColleague operation</a>	34
<a href="#">AddColleagueWithoutEmailNotification operation</a>	37
<a href="#">AddLink operation</a>	39

- [AddMembership operation](#) 41
- [AddPinnedLink operation](#) 42
- [AddSuggestions operation](#) 44
- [CreateMemberGroup operation](#) 46
- [CreateUserProfileByAccountName operation](#) 47
- [GetCommonColleagues operation](#) 49
- [GetCommonManager operation](#) 50
- [GetCommonMemberships operation](#) 52
- [GetInCommon operation](#) 54
- [GetProfileSchema operation](#) 56
- [GetProfileSchemaNameByAccountName operation](#) 57
- [GetProfileSchemaNames operation](#) 59
- [GetPropertyChoiceList operation](#) 60
- [GetUserColleagues operation](#) 62
- [GetUserLinks operation](#) 64
- [GetUserMemberships operation](#) 65
- [GetUserOrganizations operation](#) 67
- [GetUserPinnedLinks operation](#) 69
- [GetUserProfileByGuid operation](#) 71
- [GetUserProfileByIndex operation](#) 72
- [GetUserProfileByName operation](#) 75
- [GetUserProfileCount operation](#) 77
- [GetUserProfileSchema operation](#) 78
- [GetUserPropertyByAccountName operation](#) 79
- [initialization](#) 32
- [local events](#) 103
- [message processing](#) 33
- [ModifyUserPropertyByAccountName operation](#) 81
- [overview](#) 29
- [RemoveAllColleagues operation](#) 83
- [RemoveAllLinks operation](#) 84
- [RemoveAllMemberships operation](#) 86
- [RemoveAllPinnedLinks operation](#) 87
- [RemoveColleague operation](#) 89
- [RemoveLink operation](#) 90
- [RemoveMembership operation](#) 92
- [RemovePinnedLink operation](#) 94
- [sequencing rules](#) 33
- [timer events](#) 103
- [timers](#) 32
- [UpdateColleaguePrivacy operation](#) 95
- [UpdateLink operation](#) 97
- [UpdateMembershipPrivacy operation](#) 99
- [UpdatePinnedLink operation](#) 101
- [Simple types](#) 25
  - [ChoiceTypes](#) 25
  - [guid](#) 26
  - [MembershipSource](#) 26
  - [Privacy](#) 27
  - [SuggestionType](#) 27
- [Standards assignments](#) 15
- [SuggestionType simple type](#) 27
- [Synchronize two users example](#) 117

Syntax

- [messages - overview](#) 16

## T

- Timer events
  - [server](#) 103
- Timers
  - [server](#) 32
- [Tracking changes](#) 162
- [Transport](#) 16

Types

- [complex](#) 16
- [simple](#) 25

## V

- [ValueData complex type](#) 25
- [Vendor-extensible fields](#) 15
- [Versioning](#) 15

## W

- [WSDL](#) 123