WV-053 SSP – Server-Server Protocol XML Syntax Document

Version 1.2

Version 2003-Feb-21

Open Mobile Alliance
OMA-IMPS-WV-SSP_Syntax-v1_2-20030221-C

Continues the Technical Activities
Originated in the Wireless Village Initiative

Wireless Village
The Mobile IMPS Initiative

A list of errata and updates to this document is available from the Open Mobile Alliance™ Web site, http://www.openmobilealliance.org/, in the form of SIN documents, which are subject to revision or removal without notice.

This document is available online in PDF format at http://www.openmobilealliance.org/.

Known problems associated with this document are published at http://www.openmobilealliance.org/.

Comments regarding this document can be submitted to the Open Mobile Alliance™ in the manner published at
http://www.openmobilealliance.org/documents.asp

Document History	
Accepted all TRACK CHANGES in OMA-IMPS-WV-SSP_Syntax-V1_2-20021220-A.doc	2003-01-17
Updated URLs in References	2003-01-17
OMA-IMPS-WV-SSP_Syntax-v1_2-20030117-D	Draft
OMA-IMPS-WV-SSP_Syntax-v1_2-20030221-C	Current

Contents

1. Scope

The Wireless Village Instant Messaging and Presence Service (IMPS) includes four primary features:

- Presence

- Instant Messaging

- Groups

- Shared Content

Presence is the key enabling technology for IMPS. It includes client device availability (my phone is on/off, in a call), user status (available, unavailable, in a meeting), location, client device capabilities (voice, text, GPRS, multimedia) and searchable personal statuses such as mood (happy, angry) and hobbies (football, fishing, computing, dancing). Since presence information is personal, it is only made available according to the user's wishes - access control features put the control of the user presence information in the users' hands.

Instant Messaging (IM) is a familiar concept in both the mobile and desktop worlds. Desktop IM clients, two-way SMS and two-way paging are all forms of Instant Messaging. Wireless Village IM will enable interoperable mobile IM in concert with other innovative features to provide an enhanced user experience.

Groups or chat are a fun and familiar concept on the Internet. Both operators and end-users are able to create and manage groups. Users can invite their friends and family to chat in group discussions. Operators can build common interest groups where end-users can meet each other online.

Shared Content allows users and operators to setup their own storage area where they can post pictures, music and other multimedia content while enabling the sharing with other individuals and groups in an IM or chat session.

These features, taken in part or as a whole, provide the basis for innovative new services that build upon a common interoperable framework.

2. References

2.1   Normative References

[CREQ]   "Specification of WAP Conformance Requirements". Open Mobile Alliance™. WAP-221-CREQ.
                  URL:http://www1.wapforum.org/tech/terms.asp?doc=WAP-221-CREQ-20010425-a.pdf

[CSP]   "WV-042 Client-Server Protocol Session and Transactions Version 1.2". Open Mobile Alliance.
                  December 2002.
                  URL:http://www.openmobilealliance.org/member/technicalPlenary/imps/docs/OMA-IMPS-WV-
                  CSP-V1_2-20030117-D.pdf

[CSP DataType]   "WV-045 Client-Server Protocol Data Types Version 1.2". Open Mobile Alliance. December
                  2002. URL:http://www.openmobilealliance.org/member/technicalPlenary/imps/docs/OMA-IMPS-
                  WV-CSP_DataTypes-V1_2-20030117-D.pdf

[CSP SCR]   "WV-048 Client-Server Protocol Static Conformance Requirement Version 1.2". Open Mobile
                  Alliance. December 2002.
                  URL:http://www.openmobilealliance.org/member/technicalPlenary/imps/docs/OMA-IMPS-WV-
                  CSP_SCR-V1_2-20030117-D.pdf

[FeaFun]   "WV-041 Features and Functions Version 1.2". Open Mobile Alliance. December 2002.
                  URL:http://www.openmobilealliance.org/member/technicalPlenary/imps/docs/OMA-IMPS-WV-
                  Features_Functions-V1_2-20030117-D.pdf

[PA]   "WV-049 Presence Attributes Version 1.2". Open Mobile Alliance. December 2002.
                  URL:http://www.openmobilealliance.org/member/technicalPlenary/imps/docs/OMA-IMPS-WV-
                  PA-V1_2-20030117-D.pdf

[RFC2045]   "Multipurpose Internet Mail Extensions (MIME) Part one: Format of Internet Message Bodies".
                  Section 6.8 "Base64 Content-Transfer-Encoding".
                  URL:http://www.ietf.org/rfc/rfc2045.txt?number=2045

[RFC2046]   Borenstein N., and N. Freed, "MIME (Multipurpose Internet Mail Extensions) Part Two: Media
                  Types", November 1996. URL:http://www.ietf.org/rfc/rfc2046.txt?number=2046

[RFC2119]   "Keywords for using RFCs to Indicate Requirements levels", Bradner, S.
                  URL:http://www.ietf.org/rfc/rfc2119.txt?number=2119

[RFC2396]   Uniform Resource Identifiers (URI): Generic Syntax.
                  URL:http://www.ietf.org/rfc/rfc2396.txt?number=2396

[RFC2778]   "A Model for Presence and Instant Messaging", February 2000.
                  URL:http://www.ietf.org/rfc/rfc2778.txt?number=2778

[RFC822]   "Standard for the Format of ARPA Internet Text Messages", August
                  1982.URL:http://www.ietf.org/rfc/rfc0822.txt?number=822

[XML]   "Extensible Markup Language 1.0 (Second Edition)", W3C recommendation, 6-October-2000.
                  URL:http://www.w3c.org/TR/2000/REC-xml-20001006.pdf

2.2   Informative References

[Arch]   "WV-040 System Architecture Model Version 1.2". Open Mobile Alliance. December 2002.
                  URL:http://www.openmobilealliance.org/member/technicalPlenary/imps/docs/OMA-IMPS-WV-
                  Arch-V1_2-20030117-D.pdf

[FeaFun]   "WV-041 Features and Functions Version 1.2". Open Mobile Alliance. December 2002.
                      URL:http://www.openmobilealliance.org/member/technicalPlenary/imps/docs/OMA-IMPS-WV-
                      Features_Functions-V1_2-20030117-D.pdf

[CSP]   "WV-042 Client-Server Protocol Session and Transactions Version 1.2". Open Mobile Alliance.
                      December 2002.
                      URL:http://www.openmobilealliance.org/member/technicalPlenary/imps/docs/OMA-IMPS-WV-
                      CSP-V1_2-20030117-D.pdf

[CSP DTD]   "WV-043 Client-Server Protocol DTD and Examples Version 1.2". Open Mobile Alliance.
                      December 2002.
                      URL:http://www.openmobilealliance.org/member/technicalPlenary/imps/docs/OMA-IMPS-WV-
                      CSP_DTD-V1_2-20030117-D.pdf

[CSP Trans]   "WV-044 Client-Server Protocol Transport Bindings Version 1.2". Open Mobile Alliance.
                      December 2002.
                      URL:http://www.openmobilealliance.org/member/technicalPlenary/imps/docs/OMA-IMPS-WV-
                      CSP_Transport-V1_2-20030117-D.pdf

[CSP DataType]   "WV-045 Client-Server Protocol Data Types Version 1.2". Open Mobile Alliance. December
                      2002. URL:http://www.openmobilealliance.org/member/technicalPlenary/imps/docs/OMA-
                      IMPS-WV-CSP_DataTypes-V1_2-20030117-D.pdf

[CSP SMS]   "WV-046 Client-Server Protocol SMS Binding Version 1.2". Open Mobile Alliance. December
                      2002. URL:http://www.openmobilealliance.org/member/technicalPlenary/imps/docs/OMA-
                      IMPS-WV-CSP_SMS-V1_2-20030117-D.pdf

[CSP WBXML]   "WV-047 Client-Server Protocol Binary Definition and Examples Version 1.2". Open Mobile
                      Alliance. December 2002.
                      URL:http://www.openmobilealliance.org/member/technicalPlenary/imps/docs/OMA-IMPS-WV-
                      CSP_WBXML-V1_2-20030117-D.pdf

[CSP SCR]   "WV-048 Client-Server Protocol Static Conformance Requirement Version 1.2". Open Mobile
                      Alliance. December 2002.
                      URL:http://www.openmobilealliance.org/member/technicalPlenary/imps/docs/OMA-IMPS-WV-
                      CSP_SCR-V1_2-20030117-D.pdf

[PA]   "WV-049 Presence Attributes Version 1.2". Open Mobile Alliance. December 2002.
                      URL:http://www.openmobilealliance.org/member/technicalPlenary/imps/docs/OMA-IMPS-WV-
                      PA-V1_2-20030117-D.pdf

[PA DTD]   "WV-050 Presence Attribute DTD and Examples Version 1.2". Open Mobile Alliance.
                      December 2002.
                      URL:http://www.openmobilealliance.org/member/technicalPlenary/imps/docs/OMA-IMPS-WV-
                      PA_DTD-V1_2-20030117-D.pdf

[CLP]   "WV-051 Command Line Protocol Version 1.2". Open Mobile Alliance. December 2002.
                      URL:http://www.openmobilealliance.org/member/technicalPlenary/imps/docs/OMA-IMPS-WV-
                      CLP-V1_2-20030117-D.pdf

[SSP]   "WV-052 SSP - Server-Server Protocol Semantics Document Version 1.2". Open Mobile
                      Alliance. December 2002.
                      URL:http://www.openmobilealliance.org/member/technicalPlenary/imps/docs/OMA-IMPS-WV-
                      SSP-V1_2-20030117-D.pdf

[SSP Syntax]   "WV-053 Server-Server Protocol XML Syntax Document Version 1.2". Open Mobile Alliance.
                      December 2002.
                      URL:http://www.openmobilealliance.org/member/technicalPlenary/imps/docs/OMA-IMPS-WV-
                      SSP_Syntax-V1_2-20030117-D.pdf

[SSP Trans]   "WV-054 SSP - Transport Binding Version 1.2". Open Mobile Alliance. December 2002.
URL:http://www.openmobilealliance.org/member/technicalPlenary/imps/docs/OMA-IMPS-WV-SSP_Transport-V1_2-20030117-D.pdf

[SSP SCR]   "WV-055 SSP – Server-Server Protocol Static Conformance Requirement Version 1.2". Open Mobile Alliance. December 2002.
URL:http://www.openmobilealliance.org/member/technicalPlenary/imps/docs/OMA-IMPS-WV-SSP_SCR-V1_2-20030117-D.pdf

[WAPARCH]   "WAP Architecture, Version 12-July-2001". Open Mobile Alliance™. WAP-210-WAPArch.
URL:http://www1.wapforum.org/tech/terms.asp?doc=WAP-210-WAPArch-20010712-a.pdf

3. Terminology and Conventions

3.1 Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except "Scope" and "Introduction", are normative, unless they are explicitly indicated to be informative.

3.2 Definitions

None

3.3 Abbreviations

None

4. Introduction

This documents defines the valid XML documents representing a valid Wireless Village SSP protocol message using Document Type Definition mechanism. The base syntactical checking can be done by a validating XML parser. Further syntactical constraints are given in the data types assignment section.

5. Document Type Definition

```
<!--
WV Server-Server Protocol (WV SSP) V1.2 Document Type Definition

Copyright Open Mobile Alliance Ltd., 2002
          All rights reserved

WV-SSP is an XML language. Typical usage:
    <?xml version="1.0"?>
    <!DOCTYPE WV-SSP PUBLIC "-//OMA//DTD WV-SSP 1.2//EN"
              "http://www.openmobilealliance.org/DTD/WV-SSP.DTD"
              [<?oma-wv-ssp-ver supported-versions="1.2"?>]>
    <WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2">
       ...
    </WV-SSP-Message>

Terms and conditions of use are available from the
Open Mobile Alliance Ltd. web site at
http://www.openmobilealliance.org/useterms.html
-->


<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT WV-SSP-Message (SetupTransaction | Session)>
<!ATTLIST WV-SSP-Message xmlns CDATA #REQUIRED>

<!-- SetupTransaction is a transaction for session setup outside of an
   existing session -->
<!ELEMENT SetupTransaction (SendSecretToken | LoginRequest | LoginResponse)>
<!ATTLIST SetupTransaction
   mode    (Request | Response)  #REQUIRED
   transactionID CDATA      #REQUIRED>

<!-- SendSecretToken is a special request-like message without session and
   MetaInfo -->
<!ELEMENT SendSecretToken (SecretToken)>
<!ATTLIST SendSecretToken
   serviceID       CDATA   #REQUIRED
   protocol    CDATA   #FIXED  "WV-SSP"
   protocolVersion  CDATA   #FIXED  "1.2">
<!ELEMENT SecretToken (#PCDATA)>
<!ATTLIST SecretToken
   encoding     CDATA    "base64">

<!-- LoginRequest is a special response-like message without Session and
   MetaInfo -->
<!ELEMENT LoginRequest (PasswordDigest)>
<!ATTLIST LoginRequest
   serviceID       CDATA   #REQUIRED
   redirectHostId CDATA   #IMPLIED>
   timeToLive      CDATA   #IMPLIED>
<!ELEMENT PasswordDigest (#PCDATA)>
<!ATTLIST PasswordDigest
   encoding     CDATA    "base64">

<!-- LoginResponse -->
<!ELEMENT LoginResponse (Status, HostsList)>
<!ATTLIST LoginResponse
```

```
    sessionID    CDATA    #IMPLIED
    timeToLive      CDATA    #IMPLIED>
<!-- HostsList -->
<!ELEMENT HostsList (redirectHostId*)>
<!ELEMENT redirectHostId (#PCDATA)>


<!-- Status is contained in every normal response and it may also be a
    separate message or can be sent in error cases of login phase messages -->
<!ELEMENT Status (StatusDescription?)>
<!ATTLIST Status
    code CDATA    #REQUIRED>
<!ELEMENT StatusDescription (#PCDATA)>


<!-- Session -->
<!ELEMENT Session (Transaction+)>
<!ATTLIST Session
    sessionID    CDATA    #REQUIRED>

<!-- Transaction. It is a normal transaction inside an existing session -->
<!ELEMENT Transaction (Status | LogoutRequest | Disconnect |
    KeepAliveRequest | KeepAliveResponse | GetServiceRequest | ServiceList |
    ServiceNegotiation | ServiceAgreement | GetUserProfileRequest |
    UserProfile | UpdateUserProfileRequest | SearchRequest | SearchResponse |
    StopSearchRequest | InviteRequest | InviteResponse | CancelInviteRequest |
    VerifyIDRequest | VerifyIDResponse | InviteUserRequest |
    InviteUserResponse | CancelInviteUserRequest | CreateContactListRequest |
    DeleteContactListRequest | GetContactListRequest | GetContactListResponse
    | GetListMemberRequest | AddListMemberRequest | RemoveListMemberRequest |
    ContactListMemberResponse | GetListPropsRequest | SetListPropsRequest |
    ContactListPropsResponse | CreateAttrListRequest | DeleteAttrListRequest
    | GetAttrListRequest | GetAttrListResponse | AuthorizationRequest |
    AuthorizationResponse | CancelAuthRequest | GetReactiveAuthStatusRequest |
    GetReactiveAuthStatusResponse | SubscribeRequest | UnsubscribeRequest |
    SuspendPresenceNotifications | GetWatcherListRequest |
    GetWatcherListResponse | PresenceNotification | GetPresenceRequest |
    GetPresenceResponse | UpdatePresenceRequest | SendMessageRequest |
    SendMessageResponse | ForwardMessageRequest | NewMessage |
    MessageDelivered | MessageNotification | GetMessageRequest |
    SetMessageDeliveryMethod | GetMessageListRequest | GetMessageListResponse
    | RejectMessageRequest | DeliveryStatusReport | BlockUserRequest |
    GetBlockedRequest | GetBlockedResponse | CreateGroupRequest |
    DeleteGroupRequest | JoinGroupRequest | JoinGroupResponse |
    LeaveGroupRequest | LeaveGroupIndication | GetJoinedMemberRequest|
    GetJoinedMemberResponse | GetGroupMemberRequest | GetGroupMemberResponse |
    AddGroupMemberRequest | RemoveGroupMemberRequest | MemberAccessRequest |
    GetGroupPropsRequest | GetGroupPropsResponse | SetGroupPropsRequest |
    RejectListRequest | RejectListResponse | SubscribeGroupChangeRequest |
    UnsubscribeGroupChangeRequest | GetGroupSubStatusRequest |
    GetGroupSubStatusResponse | GroupChangeNotice) >
<!ATTLIST Transaction
    mode    (Request | Response)   #REQUIRED
    transactionID  CDATA       #REQUIRED>

<!-- MetaInfo is contained in every normal request -->
<!ELEMENT MetaInfo ( Requestor ) >
<!ATTLIST MetaInfo
    clientOriginated ( Yes | No )    "Yes" >
<!ELEMENT Requestor ( User? )>
```

```
<!ATTLIST Requestor
        serviceID               CDATA                   #REQUIRED>
<!ELEMENT User (ClientID?)>
<!ATTLIST User
  userID   CDATA    #REQUIRED>
<!ELEMENT ClientID EMPTY >
<!ATTLIST ClientID
  url     CDATA     #IMPLIED
  MSISDN    CDATA      #IMPLIED>


<!-- Normal PROTOCOL MESSAGES detailed. Element types used first time in the
   protocol message is defined below the protocol message element type -->


<!-- LogoutRequest -->
<!ELEMENT LogoutRequest EMPTY>


<!-- Disconnect -->
<!ELEMENT Disconnect (Status?)>


<!-- KeepAliveRequest -->
<!ELEMENT KeepAliveRequest EMPTY >
<!ATTLIST KeepAliveRequest
  timeToLive      CDATA   #IMPLIED>


<!-- KeepAliveResponse -->
<!ELEMENT KeepAliveResponse (Status)>
<!ATTLIST KeepAliveResponse
  timeToLive      CDATA   #IMPLIED>


<!-- GetServiceRequest -->
<!ELEMENT GetServiceRequest EMPTY>


<!-- ServiceList -->
<!ELEMENT ServiceList (Status?, ServiceTree)>


<!-- ServiceTree -->
<!ELEMENT ServiceTree (SRV_SAP?, SRV_Common?, SRV_Presence?, SRV_IM?,
   SRV_Group?)>


<!ELEMENT SRV_SAP (SRV_ServiceNegotiation?, SRV_UserProfileMgmt?,
   SRV_ServiceRelay?)>
<!ELEMENT SRV_ServiceNegotiation EMPTY>
<!ELEMENT SRV_UserProfileMgmt EMPTY>
<!ELEMENT SRV_ServiceRelay EMPTY>


<!ELEMENT SRV_Common (SRV_Invite?, SRV_ComplementaryInvite?, SRV_Search?,
   SRV_VerifyUser?)>
<!ELEMENT SRV_Invite (SRV_InvitePresence?, SRV_InviteSharedContent?,
   SRV_InviteIM?, SRV_InviteGroup?)>
<!ELEMENT SRV_InvitePresence EMPTY>
<!ELEMENT SRV_InviteSharedContent EMPTY>
<!ELEMENT SRV_InviteIM EMPTY>
<!ELEMENT SRV_InviteGroup EMPTY>
<!ELEMENT SRV_ComplementaryInvite EMPTY>
<!ELEMENT SRV_Search (SRV_Group?, SRV_Other?)>
<!ELEMENT SRV_Other EMPTY>
<!ELEMENT SRV_VerifyUser EMPTY>
```

```
<!ELEMENT SRV_Presence (SRV_ContactListGet?, SRV_ContactListUpdate?,
   SRV_Authorization?, SRV_WatcherList?, SRV_AttributeList?,
   SRV_ContactListAddress?, SRV_GetReactiveAuthStatus?)>
<!ELEMENT SRV_ContactListGet EMPTY>
<!ELEMENT SRV_ContactListUpdate EMPTY>
<!ELEMENT SRV_Authorization EMPTY>
<!ELEMENT SRV_WatcherList EMPTY>
<!ELEMENT SRV_AttributeList EMPTY>
<!ELEMENT SRV_ContactListAddress EMPTY>
<!ELEMENT SRV_GetReactiveAuthStatus EMPTY>

<!ELEMENT SRV_IM (SRV_SendMessage?, SRV_PushMessage?,
   SRV_MessageNotification?, SRV_GetMessage?, SRV_SetMessageDeliveryMethod?,
   SRV_GetMessageList?, SRV_RejectMessage?, SRV_DeliveryReport?,
   SRV_Blocking?, SRV_GroupHistory?)>
<!ELEMENT SRV_SendMessage (SRV_Group?, SRV_Contacts?)>
<!ELEMENT SRV_Contacts EMPTY>
<!ELEMENT SRV_PushMessage EMPTY>
<!ELEMENT SRV_MessageNotification EMPTY>
<!ELEMENT SRV_GetMessage EMPTY>
<!ELEMENT SRV_SetMessageDeliveryMethod EMPTY>
<!ELEMENT SRV_GetMessageList EMPTY>
<!ELEMENT SRV_RejectMessage EMPTY>
<!ELEMENT SRV_DeliveryReport EMPTY>
<!ELEMENT SRV_Blocking EMPTY>
<!ELEMENT SRV_GroupHistory EMPTY>

<!ELEMENT SRV_Group (SRV_GroupMGMT?, SRV_GetMember?, SRV_MemberMGMT?,
   SRV_RejectList?)>
<!ELEMENT SRV_GroupMGMT EMPTY>
<!ELEMENT SRV_GetMember EMPTY>
<!ELEMENT SRV_MemberMGMT EMPTY>
<!ELEMENT SRV_RejectList EMPTY>

<!-- ServiceNegotiation -->
<!ELEMENT ServiceNegotiation (ServiceTree) >
<!ATTLIST ServiceNegotiation
   subProtocol CDATA   #IMPLIED
   timeToLive     CDATA   #IMPLIED>

<!-- ServiceAgreement -->
<!ELEMENT ServiceAgreement (Status, ServiceTree )>
<!ATTLIST ServiceAgreement
   subProtocol CDATA   #IMPLIED
   timeToLive     CDATA   #IMPLIED>

<!-- ************************   -->
<!--      User Profile    -->
<!-- ************************   -->

<!-- GetUserProfileRequest -->
<!ELEMENT GetUserProfileRequest (MetaInfo, UserID+)>
<!ELEMENT UserID EMPTY>
<!ATTLIST UserID
   userID CDATA   #REQUIRED>

<!-- UserProfile -->
<!ELEMENT UserProfile (Status, UserProfileValue+)>
```

```
<!ELEMENT UserProfileValue (UPInfo+)>
<!ATTLIST UserProfileValue
        userID          CDATA       #REQUIRED>
<!ELEMENT UPInfo (#PCDATA)>
<!ATTLIST UPInfo
        attr            CDATA       #REQUIRED>

<!-- UpdateUserProfileRequest -->
<!ELEMENT UpdateUserProfileRequest (MetaInfo, UserProfileValue+)>

<!-- ************************   -->
<!--      GENERAL       -->
<!-- ************************   -->

<!-- SearchRequest -->
<!ELEMENT SearchRequest (MetaInfo, SearchTerm+)>
<!ATTLIST SearchRequest
        searchType    (G | U)          #IMPLIED
        searchLimit    CDATA           #IMPLIED
        searchID       CDATA           #IMPLIED
        searchIndex    CDATA           #IMPLIED>
<!ELEMENT SearchTerm EMPTY >
<!ATTLIST SearchTerm
        attr        CDATA                           #REQUIRED
        value       CDATA                           #REQUIRED>

<!-- SearchResponse -->
<!ELEMENT SearchResponse (Status, SearchResult?)>
<!ATTLIST SearchResponse
      searchID         CDATA  #IMPLIED
      searchFindings   CDATA  #REQUIRED
      searchIndex      CDATA  #REQUIRED
      completed       ( Yes | No ) #REQUIRED>
<!ELEMENT SearchResult ( (User | ScreenName)* | GroupID+ )>
<!ELEMENT GroupID EMPTY >
<!ATTLIST GroupID
        groupID   CDATA      #REQUIRED>
<!ELEMENT ScreenName (#PCDATA) >
<!ATTLIST ScreenName
        groupID   CDATA      #REQUIRED>

<!-- StopSearchRequest -->
<!ELEMENT StopSearchRequest (MetaInfo)>
<!ATTLIST StopSearchRequest
      searchID         CDATA  #REQUIRED>

<!-- InviteRequest -->
<!ELEMENT InviteRequest (MetaInfo, Inviting, Invited, GroupID?,
  AttributeList?, ContentIDList?, InviteNote?)>
<!ATTLIST InviteRequest
  inviteID    CDATA      #REQUIRED
  inviteType     (GR | IM | PR | SC | GM) #REQUIRED
  validity    CDATA      #IMPLIED>
<!ELEMENT Inviting (User | ScreenName)>
<!ELEMENT Invited (User | ScreenName | ContactListID | GroupID)>
<!ELEMENT ContactListID EMPTY>
<!ATTLIST ContactListID
  contactListID CDATA      #REQUIRED>
```

```
<!ELEMENT AttributeList (PresenceSubList)>


<!ELEMENT ContentIDList (ContentID+)>
<!ELEMENT ContentID EMPTY >
<!ATTLIST ContentID
  url CDATA   #REQUIRED>
<!ELEMENT InviteNote (#PCDATA)>

<!-- InviteResponse -->
<!ELEMENT InviteResponse (Status, Inviting, Responding, ResponseNote?)>
<!ATTLIST InviteResponse
  inviteID    CDATA      #REQUIRED
acceptance     ( Yes | No ) #REQUIRED>
<!ELEMENT Responding (User | ScreenName | GroupID)>
<!ELEMENT ResponseNote (#PCDATA)>

<!-- InviteUserRequest -->
<!ELEMENT InviteUserRequest (MetaInfo, Inviting, Invited, GroupID?,
  AttributeList?, ContentIDList?, InviteNote?)>
<!ATTLIST InviteUserRequest
  inviteID    CDATA      #REQUIRED
  inviteType     (GR | IM | PR | SC | GM) #REQUIRED
  validity    CDATA      #IMPLIED>

<!-- InviteUserResponse -->
<!ELEMENT InviteUserResponse (Status, Inviting, Responding, ResponseNote?)>
<!ATTLIST InviteUserResponse
  inviteID    CDATA      #REQUIRED
acceptance     ( Yes | No ) #REQUIRED>

<!-- CancelInviteRequest -->
<!ELEMENT CancelInviteRequest (MetaInfo, Canceling, Canceled?,
  ContentIDList?, CancelNote?)>
<!ATTLIST CancelInviteRequest
  inviteID    CDATA   #REQUIRED>
<!ELEMENT Canceling (User | ScreenName)>
<!ELEMENT Canceled (User | GroupID | ContactListID)>
<!ELEMENT CancelNote (#PCDATA)>

<!-- CancelInviteUserRequest -->
<!ELEMENT CancelInviteUserRequest (MetaInfo, Canceling, Canceled,
  ContentIDList?, CancelNote?)>
<!ATTLIST CancelInviteUserRequest
  inviteID    CDATA   #REQUIRED>

<!-- VerifyIDRequest -->
<!ELEMENT VerifyIDRequest (MetaInfo, WVIDList)>
<!ELEMENT WVIDList (VerifyUserID*, VerifyContactListID*, VerifyGroupID*,
  VerifyScreenName*, VerifyDomain*)>
<!ELEMENT VerifyUserID (DateTime?)>
<!ATTLIST VerifyUserID
  userID CDATA   #REQUIRED>
<!ELEMENT VerifyContactListID (DateTime?)>
<!ATTLIST VerifyContactListID
  contactListID CDATA   #REQUIRED>
<!ELEMENT VerifyGroupID (DateTime?)>
<!ATTLIST VerifyGroupID
```

```
      groupIDCDATA   #REQUIRED>
<!ELEMENT VerifyScreenName (DateTime?)>
<!ATTLIST VerifyScreenName
   screenName  CDATA   #REQUIRED>
<!ELEMENT VerifyDomain (DateTime?)>
<!ATTLIST VerifyDomain
   domain CDATA   #REQUIRED>
<!ELEMENT DateTime (#PCDATA)>
<!ATTLIST DateTime
   format    CDATA      "iso8601">


<!-- VerifyIDResponse -->
<!ELEMENT VerifyIDResponse (Status, WVIDList)>


<!-- ************************   -->
<!--      CONTACT LIST    -->
<!-- ************************   -->


<!—- CreateContactListRequest -->
<!ELEMENT CreateContactListRequest (MetaInfo, ContactUser*)>
<!ATTLIST CreateContactListRequest
   contactListID CDATA   #REQUIRED>
<!ELEMENT ContactUser EMPTY >
<!ATTLIST ContactUser
   userID    CDATA       #REQUIRED
   nick    CDATA     #IMPLIED>


<!—- DeleteContactListRequest -->
<!ELEMENT DeleteContactListRequest (MetaInfo, ContactListID+)>


<!—- GetContactListRequest -->
<!ELEMENT GetContactListRequest (MetaInfo, Version)>
<!ELEMENT Version (#PCDATA)>
<!ATTLIST Version
   format    CDATA      "DateTime">


<!—- GetContactListResponse -->
<!ELEMENT GetContactListResponse (MetaInfo, Version, ContactListID*)>
<!ATTLIST GetListMemberRequest
   defaultContactListID    CDATA      #IMPLIED>


<!—- GetListMemberRequest -->
<!ELEMENT GetListMemberRequest (MetaInfo)>
<!ATTLIST GetListMemberRequest
   contactListID    CDATA      #REQUIRED>


<!—- AddListMemberRequest -->
<!ELEMENT AddListMemberRequest (MetaInfo, ContactUser+)>
<!ATTLIST AddListMemberRequest
   contactListID    CDATA      #REQUIRED>


<!—- RemoveListMemberRequest -->
<!ELEMENT RemoveListMemberRequest (MetaInfo, ContactUserSpec+)>
<!ATTLIST RemoveListMemberRequest
   contactListID CDATA   #REQUIRED>
<!ELEMENT ContactUserSpec EMPTY >
<!ATTLIST ContactUserSpec
   userID    CDATA   #IMPLIED
```

```
     nick      CDATA   #IMPLIED>

 <!—- ContactListMemberResponse -->
 <!ELEMENT ContactListMemberResponse (Status, ContactUser*)>
 <!ATTLIST ContactListMemberResponse
   contactListID    CDATA   #REQUIRED>

 <!—- GetListPropsRequest -->
 <!ELEMENT GetListPropsRequest (MetaInfo)>
 <!ATTLIST GetListPropsRequest
   contactListID    CDATA   #REQUIRED>

 <!—- SetListPropsRequest -->
 <!ELEMENT SetListPropsRequest (MetaInfo, ContactListProperties)>
 <!ATTLIST SetListPropsRequest
   contactListID CDATA   #REQUIRED>
 <!ELEMENT ContactListProperties (Property*)>
 <!ELEMENT Property (#PCDATA)>
 <!ATTLIST Property
   prop      CDATA     #REQUIRED>

 <!—- ContactListPropsResponse -->
 <!ELEMENT ContactListPropsResponse (Status, ContactListProperties)>

 <!—- CreateAttrListRequest -->
 <!ELEMENT CreateAttrListRequest (MetaInfo, AttributeList, ContactListID*,
   UserID*)>
 <!ATTLIST CreateAttrListRequest
   defaultList ( Yes | No )   #REQUIRED>

 <!—- DeleteAttrListRequest -->
 <!ELEMENT DeleteAttrListRequest (MetaInfo, ContactListID*, UserID*)>
 <!ATTLIST DeleteAttrListRequest
   defaultList ( Yes | No )   #REQUIRED>

 <!—- GetAttrListRequest -->
 <!ELEMENT GetAttrListRequest (MetaInfo, AttributeList, ContactListID*,
   UserID*)>
 <!ATTLIST GetAttrListRequest
   defaultList    ( Yes | No )   #REQUIRED
   exportList     ( Yes | No )   #REQUIRED>

 <!—- GetAttrListResponse -->
 <!ELEMENT GetAttrListResponse (Status, AttributeAssociation*,
   DefaultAttributeList?)>
 <!ATTLIST GetAttrListResponse
   defaultList    ( Yes | No )   #REQUIRED
   exportList     ( Yes | No )   #REQUIRED>
 <!ELEMENT AttributeAssociation ( (UserID | ContactListID), AttributeList)>
 <!ELEMENT DefaultAttributeList (PresenceSubList )>

 <!-- ************************    -->
 <!--      PRESENCE   -->
 <!-- ************************    -->

 <!-- SubscribeRequest -->
 <!ELEMENT SubscribeRequest (MetaInfo, UserID*, ContactListID*, AttributeList?
   , AutoSubscribe)>
```

```
<!ELEMENT AutoSubscribe (#PCDATA)>

<!-- AuthorizationRequest -->
<!ELEMENT AuthorizationRequest (MetaInfo, AuthRequestTuple+)>
<!ELEMENT AuthRequestTuple (Subscribers, Authorizer, AttributeList?)>
<!ATTLIST AuthRequestTuple
  authID    CDATA      #REQUIRED>
<!ELEMENT Authorizer EMPTY>
<!ATTLIST Authorizer
  userID    CDATA      #REQUIRED>
<!ELEMENT Subscribers (UserID*)>

<!-- AuthorizationResponse -->
<!ELEMENT AuthorizationResponse (Status, AuthResponseTuple+)>
<!ELEMENT AuthResponseTuple (Authorizer, SubscriberResult+)>
<!ATTLIST AuthResponseTuple
  authID    CDATA      #REQUIRED>
<!ELEMENT SubscriberResult (UserID, PresenceSubList?)>
<!ATTLIST SubscriberResult
  granted   CDATA      #REQUIRED>

<!-- CancelAuthRequest -->
<!ELEMENT CancelAuthRequest (MetaInfo, UserID*)>

<!-- GetReactiveAuthStatusRequest -->
<!ELEMENT GetReactiveAuthStatusRequest (MetaInfo, UserID*)>
<!ELEMENT GetReactiveAuthStatusResponse (ReactiveAuthStatusList)>
<!ELEMENT ReactiveAuthStatusList (ReactiveAuthStatus*)>
<!ELEMENT ReactiveAuthStatus (UserID, ReactiveAuthState, PresenceSubList?)>
<!ELEMENT ReactiveAuthState (#PCDATA)>

<!-- UnsubscribeRequest -->
<!ELEMENT UnsubscribeRequest (MetaInfo, UserID*, ContactListID*)>
<!-- AttributeList? -->

<!—- SuspendPresenceNotifications -->
<!ELEMENT SuspendPresenceNotifications (MetaInfo, UserID*, ContactListID*)>

<!-- PresenceNotification -->
<!ELEMENT PresenceNotification (MetaInfo, Subscribers, PresenceValue+,
  Version?)>
<!ELEMENT PresenceValue ( PresenceSubList, Version? )>
<!ATTLIST PresenceValue
         userID           CDATA      #REQUIRED>

<!ELEMENT PresenceSubList (#PCDATA)>
<!ATTLIST PresenceSubList
      xmlns     CDATA      #REQUIRED
      xmlns:Ext CDATA      #IMPLIED>


<!-- GetWatcherListRequest -->
<!ELEMENT GetWatcherListRequest (MetaInfo)>

<!-- GetWatcherListResponse -->
<!ELEMENT GetWatcherListResponse (Status, UserID*)>

<!-- GetPresenceRequest -->
```

```
<!ELEMENT GetPresenceRequest (MetaInfo, (VerUserID | VerContactListID)+,
  AttributeList)>
<!ELEMENT VerUserID (Version?)>
<!ATTLIST VerUserID
  userID CDATA   #REQUIRED>
<!ELEMENT VerContactListID (Version?)>
<!ATTLIST VerContactListID
  userID CDATA   #REQUIRED>

<!-- GetPresenceResponse -->
<!ELEMENT GetPresenceResponse (Status, PresenceValue*, Version?)>

<!-- UpdatePresenceRequest -->
<!ELEMENT UpdatePresenceRequest (MetaInfo, PresenceValue+ )>

<!-- *************************   -->
<!--   INSTANT MESSAGE    -->
<!-- *************************   -->

<!-- SendMessageRequest -->
<!ELEMENT SendMessageRequest (MetaInfo, MessageInfo, ContentData)>
<!ATTLIST SendMessageRequest
  deliveryReport   ( Yes | No )   #REQUIRED >
<!ELEMENT MessageInfo (Recipient+, Sender, DateTime)>
<!ATTLIST MessageInfo
  messageID       CDATA     #IMPLIED
  messageURI      CDATA     #IMPLIED
  contentType     CDATA     #IMPLIED
  contentSize     CDATA     #IMPLIED
  validity        CDATA     #IMPLIED>
<!ELEMENT Recipient ((UserID | ScreenName | GroupID | ContactListID),
  RecipientDisplay?)>
<!ELEMENT Sender ((UserID | GroupID), SenderDisplay?)>
<!ELEMENT RecipientDisplay (UserID | ScreenName | GroupID | Name)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT SenderDisplay (UserID | ScreenName | GroupID | Name)>
<!ELEMENT ContentData (#PCDATA)>
<!ATTLIST ContentData
contentType    CDATA     #REQUIRED
encoding       CDATA     "base64">

<!-- SendMessageResponse -->
<!ELEMENT SendMessageResponse (Status)>
<!ATTLIST SendMessageResponse
  messageID   CDATA     #REQUIRED>

<!-- ForwardMessageRequest -->
<!ELEMENT ForwardMessageRequest (MetaInfo, Recipient+)>
<!ATTLIST ForwardMessageRequest
  messageID      CDATA     #IMPLIED
  messageURI     CDATA     #IMPLIED>

<!-- NewMessage -->
<!ELEMENT NewMessage ((MetaInfo | Status), RecipientIDs, MessageInfo,
  ContentData)>
<!ATTLIST NewMessage
  messageID      CDATA     #IMPLIED
  messageURI     CDATA     #IMPLIED>
```

```
<!ELEMENT RecipientIDs (UserID+)>

<!-- MessageDelivered -->
<!ELEMENT MessageDelivered ( MetaInfo | Status )>
<!ATTLIST MessageDelivered
  messageID   CDATA     #REQUIRED>

<!-- MessageNotification -->
<!ELEMENT MessageNotification ( MetaInfo, MessageInfo, RecipientIDs )>
<!ATTLIST MessageNotification
  messageID   CDATA     #REQUIRED>

<!-- GetMessageRequest -->
<!ELEMENT GetMessageRequest ( MetaInfo )>
<!ATTLIST GetMessageRequest
  messageID   CDATA     #REQUIRED>

<!-- SetMessageDeliveryMethod -->
<!ELEMENT SetMessageDeliveryMethod ( MetaInfo )>
<!ATTLIST SetMessageDeliveryMethod
  messageID       CDATA   #REQUIRED
  deliveryMethod      (NotifyGet | Push)  #REQUIRED
  acceptedContentLength   CDATA    #REQUIRED
  groupID     CDATA   #IMPLIED>

<!-- GetMessageListRequest -->
<!ELEMENT GetMessageListRequest ( MetaInfo )>
<!ATTLIST GetMessageListRequest
  groupID     CDATA   #IMPLIED
  messageCount  CDATA   #IMPLIED>

<!-- GetMessageListResponse -->
<!ELEMENT GetMessageListResponse ( Status, MessageInfo )>

<!-- RejectMessageRequest -->
<!ELEMENT RejectMessageRequest ( MetaInfo, MessageSpec* )>
<!ELEMENT MessageSpec EMPTY >
<!ATTLIST MessageSpec
  messageID   CDATA     #IMPLIED
  messageURI    CDATA     #IMPLIED>

<!-- DeliveryStatusReport -->
<!ELEMENT DeliveryStatusReport (MetaInfo, DeliveryResult, DeliveryTime?,
  MessageInfo)>
<!ELEMENT DeliveryResult (Status)>
<!ELEMENT DeliveryTime (#PCDATA)>

<!-- BlockUserRequest -->
<!ELEMENT BlockUserRequest ( MetaInfo, BlockList?, UnblockList?, GrantList?,
  UngrantList? )>
<!ATTLIST BlockUserRequest
  blockListStatus  (Active | Inactive)    #REQUIRED
  grantListStatus  (Active | Inactive) #REQUIRED>
<!ELEMENT BlockList (UserID*, ScreenName*, GroupID*)>
<!ELEMENT UnblockList (UserID*, ScreenName*, GroupID*)>
<!ELEMENT GrantList (UserID*, ScreenName*, GroupID*)>
<!ELEMENT UngrantList (UserID*, ScreenName*, GroupID*)>
```

```
<!-- GetBlockedRequest -->
<!ELEMENT GetBlockedRequest ( MetaInfo )>

<!-- GetBlockedResponse -->
<!ELEMENT GetBlockedResponse ( Status, BlockList, GrantList )>
<!ATTLIST GetBlockedResponse
  blockListStatus  (Active | Inactive)    #REQUIRED
  grantListStatus  (Active | Inactive)    #REQUIRED>

<!-- *************************   -->
<!--         GROUP        -->
<!-- *************************   -->

<!-- CreateGroupRequest -->
<!ELEMENT CreateGroupRequest (MetaInfo, GroupProperties, ScreenName?)>
<!ATTLIST CreateGroupRequest
  groupID   CDATA     #REQUIRED
  joinGroup (Yes | No) #REQUIRED
  subscribeNotif (Yes | No)   #REQUIRED>
<!ELEMENT GroupProperties (Property+, WelcomeNote?)>
<!ELEMENT WelcomeNote (#PCDATA)>

<!-- DeleteGroupRequest -->
<!ELEMENT DeleteGroupRequest (MetaInfo)>
<!ATTLIST DeleteGroupRequest
  groupID   CDATA     #REQUIRED>

<!-- JoinGroupRequest -->
<!ELEMENT JoinGroupRequest (MetaInfo, ScreenName?, OwnProperties?)>
<!ATTLIST JoinGroupRequest
  groupID   CDATA       #REQUIRED
  joinedListRequest( Yes | No ) #REQUIRED
  subscribeNotif (Yes | No)     #REQUIRED>

<!-- JoinGroupResponse -->
<!ELEMENT JoinGroupResponse (Status, JoinedList, WelcomeNote?)>
<!ELEMENT JoinedList (Name*)>

<!-- LeaveGroupRequest -->
<!ELEMENT LeaveGroupRequest (MetaInfo, GroupID)>
<!ATTLIST LeaveGroupRequest
  groupID   CDATA     #REQUIRED>

<!-- LeaveGroupIndication -->
<!ELEMENT LeaveGroupIndication ((MetaInfo | Status), ReasonText)>
<!ELEMENT ReasonText (#PCDATA)>
<!ATTLIST LeaveGroupIndication
  groupID   CDATA     #REQUIRED>

<!-- GetJoinedMemberRequest -->
<!ELEMENT GetJoinedMemberRequest (MetaInfo)>
<!ATTLIST GetJoinedMemberRequest
  groupID   CDATA     #REQUIRED>

<!-- GetJoinedMemberResponse -->
<!ELEMENT GetJoinedMemberResponse (Status, JoinedUser*)>
<!ELEMENT JoinedUser (#PCDATA)>
<!ATTLIST JoinedUser
```

```
     userID     CDATA      #REQUIRED>

<!-- GetGroupMemberRequest -->
<!ELEMENT GetGroupMemberRequest (MetaInfo)>
<!ATTLIST GetGroupMemberRequest
   groupID   CDATA      #REQUIRED>

<!-- GetGroupMemberResponse -->
<!ELEMENT GetGroupMemberResponse (Status, Admins?, Moderators?,
   OrdinaryUsers?)>
<!ELEMENT Admins (UserID*)>
<!ELEMENT Moderators (UserID*)>
<!ELEMENT OrdinaryUsers (UserID*)>

<!-- AddGroupMemberRequest -->
<!ELEMENT AddGroupMemberRequest (MetaInfo, UserID*)>
<!ATTLIST AddGroupMemberRequest
   groupID   CDATA      #REQUIRED>

<!-- RemoveGroupMemberRequest -->
<!ELEMENT RemoveGroupMemberRequest (MetaInfo, UserID+)>
<!ATTLIST RemoveGroupMemberRequest
   groupID   CDATA      #REQUIRED>

<!-- MemberAccessRequest -->
<!ELEMENT MemberAccessRequest (MetaInfo, Admins?, Moderators?,
   OrdinaryUsers?)>
<!ATTLIST MemberAccessRequest
   groupID   CDATA      #REQUIRED>

<!-- GetGroupPropsRequest -->
<!ELEMENT GetGroupPropsRequest (MetaInfo)>
<!ATTLIST GetGroupPropsRequest
   groupID   CDATA      #REQUIRED>

<!-- GetGroupPropsResponse -->
<!ELEMENT GetGroupPropsResponse (Status, GroupProperties, OwnProperties)>
<!ELEMENT OwnProperties (Property+)>

<!-- SetGroupPropsRequest -->
<!ELEMENT SetGroupPropsRequest (GroupProperties?, OwnProperties?)>
<!ATTLIST SetGroupPropsRequest
   groupID   CDATA      #REQUIRED>

<!-- RejectListRequest -->
<!ELEMENT RejectListRequest (MetaInfo, AddUsers, RemoveUsers)>
<!ATTLIST RejectListRequest
   groupID   CDATA      #REQUIRED>
<!ELEMENT AddUsers (UserID*)>
<!ELEMENT RemoveUsers (UserID*)>

<!-- RejectListResponse -->
<!ELEMENT RejectListResponse (Status, RejectList)>
<!ELEMENT RejectList (UserID*)>

<!-- SubscribeGroupChangeRequest -->
<!ELEMENT SubscribeGroupChangeRequest (MetaInfo)>
<!ATTLIST SubscribeGroupChangeRequest
```

```
    groupID   CDATA     #REQUIRED>

<!-- UnsubscribeGroupChangeRequest -->
<!ELEMENT UnsubscribeGroupChangeRequest (MetaInfo)>
<!ATTLIST UnsubscribeGroupChangeRequest
   groupID   CDATA     #REQUIRED>

<!-- GetGroupSubStatusRequest -->
<!ELEMENT GetGroupSubStatusRequest (MetaInfo)>
<!ATTLIST GetGroupSubStatusRequest
   groupID   CDATA     #REQUIRED>

<!-- GetGroupSubStatusResponse -->
<!ELEMENT GetGroupSubStatusResponse (Status)>
<!ATTLIST GetGroupSubStatusResponse
   groupID   CDATA     #REQUIRED
   subscribed    (Yes | No)     #REQUIRED>

<!-- GroupChangeNotice -->
<!ELEMENT GroupChangeNotice (Subscribers, Joined?, Left?, GroupProperties?,
   OwnProperties?)>
<!ATTLIST GroupChangeNotice
   groupID   CDATA     #REQUIRED>
<!ELEMENT Joined (Name*)>
<!ELEMENT Left (Name*)>
```

6.   Data Type Assignments for SSP XML ELEMENTS

This section provides the mapping of the informational elements in SSP primitives to XML element constructions or attributes. If the mapped value is an element construction, then the specified element construction inside the transaction element represents the SSP information element. If the mapped value is an attribute, then the information element is represented by the specified attribute of the container element (the transaction element).

Information Element	XML Element	XML Attribute
Accepted-Content-Length		acceptedContentLength
Add-User-ID-List	AddUsers	
Agreed-Service-List	ServiceTree	
Agreed-Sub-Protocol		subProtocol
Agreed-Time-To-Live		timeToLive
Attribute-Association-List	AttributeAssociation+	
Authorization-Result	SubscriberResult+	
Authorizing-User-ID	Authorizer	
Auto-Subscribe	AutoSubscribe	
Block-List-Status		blockListStatus
Block-Entity-List	BlockList	
Canceled-Content-ID-List	ContentIDList	
Canceled-User-ID-List	UserID+	
Canceled-User	Canceled	
Canceling-User	Canceling	
Cancel-Reason	CancelNote	
Client-ID		clientID
Client-Originated		clientOriginated
Completed		completed
Contact-List-ID		contactListID
Contact-List-ID-List	ContactListID+	
Contact-List-ID-List of GetPresenceRequest	VerContactListID+	
Contact-List-Props	ContactListProperties	
Contact-List-Version	Version	
Content	ContentData	

Deafult-Association-List	DefaultAttributeList	
Default-C-List-ID		defaultContactListID
Default-List		defaultList
Delivery-Result	DeliveryReport	
Delivery-Report-Request		deliveryReport
Delivery-Time	DeliveryTime	
Desired-Service-List	ServiceTree	
Desired-Sub-Protocol		subProtocol
Export-List		exportList
Grant-List-Status		grantListStatus
Grant-Entity-List	GrantList	
Group-ID		groupID
Group-Prop-List	GroupProperties	
Invite-Acceptance		acceptance
Invite-Content-ID-List	ContentIDList	
Invite-Group-ID	GroupID	
Invite-ID		inviteID
Invite-Presence-Attribute-List	AttributeList	
Invite-Reason	InviteNote	
Invite-Response	ResponseNote	
Invite-Type		inviteType
Invited-User	Invited	
Inviting-User	Inviting	
Join-Group		joinGroup
Joined-Request		joinedListRequest
Joined-User-Screen-Name-List	JoinedUser+	
Joined-User-Screen-Name-List of GroupChangeNotice	Joined	
Left-User-Screen-Name-List	Left	
List-of-Hosts	HostsList	
List-of-Subscribing-User-ID-and-Presence-Attribute-List	AuthRequestTuple/Subscribers and AuthRequestTuple/AttributeL	

	ist	
Message-Count		messageCount
Message-Delivery-Method		deliveryMethod
Message-ID		messageID
Message-ID-List		messageID of MessageSpec+
Message-Info	MessageInfo	
Message-URI		messageURI
Message-URI-List		messageID of MessageSpec+
Meta-Information	MetaInfo	
Own-Prop-List	OwnProperties	
Password-Digest	PasswordDigest	
Presence-Attribute-List	AttributeList	
Presence-Value-List	PresenceValue+	
Presence-Value-Version	Version	
Presence-Value-Version-List of GetPresenceRequest	Version of (VerUserID and VerContactListID	
Presence-Value-Version-List	Version of (PresenceValue+)	
Protocol		protocol
Protocol-Version		protocolVersion
Reason-Text	ReasonText	
Recipients	Recipient+	
Recipient-UserID-List	RecipientIDs	
Redirect-HostID		redirectHostId
Reject-User-ID-List	RejectList	
Remove-User-ID-List	RemoveUsers	
Responding-User	Responding	
Screen-Name	ScreenName	
Search-Criteria	SearchCriteria	
Search-Findings		searchFindings
Search-ID		searchID
Search-Index		searchIndex

Search-Limit		searchLimit
Search-Pair-List	SearchTerm+	
Search-Results	SearchResult	
Search-Type		searchType
SecretToken	SecretToken	
Service-ID		serviceID
Service-List	ServiceTree	
Session-ID		sessionID
Status code		code
Status description		statusDescription
Status-Info	Status	
Subscribe-Notif		subscribeNotif
Subscribing-User-ID-List	Subscribers	
Subscribing-User-ID-List in AuthorizationResponse	SubscriberResult+	
Subscription-Status		subscribed
Time-To-Live		timeToLive
Transaction-ID		transactionID
Unblock-Entity-List	UnblockList	
Ungrant-Entity-List	UngrantList	
Updated-User-Profile-List	UserProfileValue+	
User-ID		userID
User-ID-List	UserID+	
User-ID-List of GetPresenceRequest	VerUserID+	
User-ID-List of (GetGroupMemberResponse or MemberAccessRequest)	OrdinaryUsers	
User-ID-List-Adm	Admins	
User-ID-List-Mod	Moderators	
User-List	ContactUser+	
User-List in RemoveListMemberRequest	ContactUserSpec+	
User-Profile-List	UserProfileValue+	

Validity		validity
Welcome -Text	WelcomeNote	
WV-ID-List	WVIDList	

7. Data Types for XML Elements and attributes

7.1 Redefined and new data types

7.1.1 Boolean

A boolean value indicates either true or false.

It is encoded to XML as an enumerated type with possible values of:

- "Yes" : means the specified value is true.
- "No"   : means the specified value is false.

These values are case sensitive. Other values are invalid.

7.1.2 MSISDN

MSISDN is a string containing a valid MSISDN number.

7.1.3 URL

The value of the type is a string of a Uniform Resource Locator. See rfc2396.

7.2 Data types for XML Attributes

The XML attributes of the elements used are defined in this section. Description of the basic types can be found in [SSP].

XML Attribute	acceptance
Data type	Boolean
Description	Specifies if the invited party accepted the invitation.

XML Attribute	acceptedContentLength
Data type	Integer
Description	Maximum size of message that can be pushed to the user. See [SSP].

XML Attribute	authID
Data type	String
Description	Gives an identifier for an authentication request tuple. It is specified by the requestor server. Unique for a session. Case sensitive.

XML Attribute	blockListStatus
Data type	Enum
Range	"Active" | "Inactive"
Description	Indicates if the block list is in use ("Active") or not ("Inactive"). See [SSP].

XML Attribute	clientOriginated
Data type	Boolean
Description	Attribute of the Transaction method. It specifies whether the transaction is originated from a client (Yes) or from a Wireless Village server (No).

XML Attribute	code
Data type	Integer
Description	Defined only in the Status element. It specifies the status code. Possible status codes are defined in the [SSP].

XML Attribute	completed
Data type	Boolean
Description	Indicates the completion of the search on the server. 'No' if still searching and new results may come, 'Yes' if server has finished searching, and new results will not come anymore.

XML Attribute	contactListID
Data type	String
Description	Corresponds to the Contact-List-ID defined in [SSP]. Case insensitive.

XML Attribute	contentType
Data type	String
Description	Specifies the content type of an instant message. Must be in MIME type format.

XML Attribute	defaultList
Data type	Boolean
Description	Specifies if the container attribute list operation element is for the default list of the user ("Yes") or for an ordinary list ("No").

XML Attribute	deliveryMethod
Data type	Enum
Range	"NotifyGet" | "Push"
Description	Specifies the method used for message delivery. See [SSP] and [CSP].

XML Attribute	deliveryReport
Data type	Boolean
Description	Specifies if a delivery report is requested for the sended instant message.

XML Attribute	encoding
Data type	String
Description	Defines the used encoding method for the content of the element. Encoding scheme is base64 by default (i.e. when this attribute is not specified).

XML Attribute	exportList
Data type	Boolean
Description	See [SSP].

XML Attribute	format
Data type	String
Description	Defines the format of the container element. In this version this value is only specified for Version, for what the default format is DateTime. This format means a value in DateTime format specified in [SSP].

XML Attribute	granted
Data type	Boolean
Description	Specifies for a user if his authentication request was allowed ("Yes") or not ("No").

XML Attribute	grantListStatus
Data type	Enum
Range	"Active"	"Inactive"
Description	Indicates if the grant list is in use ("Active") or not ("Inactive"). See [SSP].

XML Attribute	groupID
Data type	String
Description	Corresponds to the Group-ID defined in [SSP]. Case insensitive.

XML Attribute	inviteID
Data type	String
Description	See in [SSP]. Case sensitive.

XML Attribute	inviteType
Data type	Enum
Range	"GR"	"IM"	"PR"	"SC"	"GM"
Description	Defines the type of the invitation. In case of value "GR", the invitation is for a group, in case of value "IM", the invitation is for instant messaging, in case of "PR", the invitation is for a presence attribute, in case of "SC" the invitation is for a shared content. In case of "GM" the invitation is for group membership.

XML Attribute	joinGroup
Data type	Boolean
Description	Indicates that the newly created group is joined (or not) at creation time.

XML Attribute	joinedListRequest
Data type	Boolean
Description	See in [SSP].

XML Attribute	messageCount
Data type	Integer
Description	The maximum number of message-info structures to be returned.

XML Attribute	messageID
Data type	String
Description	Defines a message identifier that uniquely identifies the message. The generator of the message must guarantee that the message identifier is globally unique. Must be in format msg-id of rfc2822. For the right-hand side of the msg-id, the domain name of the server should be used. Case sensitive.

XML Attribute	messageURI
Data type	URI
Description	Specifies a message by a Uniform Resource Locator.

XML Attribute	mode
Data type	Integer
Range	"Request"	"Response"
Description	Attribute of the Transaction method. It specifies whether the transaction is a request of a response without parsing of the content of the transaction.

XML Attribute	MSISDN
Data type	MSISDN
Description	Specifies a valid phone number, by which for example a client can be identified.

XML Attribute	nick
Data type	String
Description	Defines a nickname for a user on the contact list.

XML Attribute	prop
Data type	String
Description	Defines a property name.

XML Attribute	protocol
Data type	String
Range	"WV-SSP"
Description	Defines the procotol used. Must be "WV-SSP".

XML Attribute	protocolVersion
Data type	String
Range	"1.2"
Description	Defines the version of the procotol used. Must be "1.2".

XML Attribute	redirectHostId
Data type	String
Description	Redirect host DNS name

XML Attribute	searchFindings
Data type	Integer
Description	See in [SSP].

XML Attribute	searchID
Data type	String
Description	See in [SSP]. Case sensitive.

XML Attribute	searchIndex
Data type	Integer
Description	See in [SSP].

XML Attribute	searchLimit
Data type	Integer
Description	See in [SSP].

XML Attribute	searchType
Data type	Enum
Range	"G"	"U"
Description	Defines the type of the search. In case of value "G", the search is for a group, in case of value "U", the search is for users.

XML Attribute	serviceID
Data type	String
Format	Defined in [SSP].
Description	See [SSP]. Case sensitive.

XML Attribute	sessionID
Data type	String
Description	Specifies the used session identifier. Case sensitive.

XML Attribute	subscribed
Data type	Boolean
Description	Used in a subscription status response. Specifies if the asked entity is subscribed ("Yes") to the group or not ("No").

XML Attribute	subscribeNotif
Data type	Boolean
Description	A flag indicating that the user wants to activate the group change notifications while joining the group.

XML Attribute	subProtocol
Data type	String
Description	During session negotiation it specifies the subprotocol name and version for proprietary extensions.

XML Attribute	timeToLive
Data type	Integer
Description	Defines the time in second how long the containing action is valid.

XML Attribute	url
Data type	URL
Description	Identifies a content or a client by a Uniform Resource Locator.

XML Attribute	userID
Data type	String
Description	Corresponds to the User-ID defined in [SSP]. Case insensitive.

XML Attribute	validity
Data type	Integer
Description	Specifies the time in seconds how long the specified entity is regarded valid.

XML Attribute	value
Data type	String
Description	Specifies the value for the attribute and the operand of the search to be executed.

XML Attribute	transactionID
Data type	String
Description	Identifies a SSP transaction. The initiating party assigns this ID. Case sensitive.

XML Attribute	xmlns
Data type	URL
Description	Specifies the namespace the message is using.

XML Attribute	Data Types for XML Elements and attributes
Data type	Boolean
Description	A flag indicating that the user wants to activate the group change notifications while joining the group.

7.3 Data types for XML Terminal Elements

This section will define the SSP specific or redefined XML elements. Description of the basic types and description of terminal elements common with CSP can be found in [CSP DataType]. If an XML terminal element definition is found in both this document and in [CSP DataType] then the given XML terminal elements is redefined by this document and the definition below must be used. Empty terminal elements representing protocol primitives are not listed here.

For description of the XML terminal elements inside the ServiceTree element see [SSP] document.

XML ELEMENT	Authorizer
Data type	EMPTY
Format	
Description	Defines an authorizer user using userid.
Range	

Table 1. Authorizer

XML ELEMENT	AutoSubscribe
Data type	Boolean
Format	Defined in section 7.1.1
Description	'Yes' means that the automatic subscription to the presence attributes is enabled when a new user is added to the contact list, and the automatic unsubscription to the presence attributes is also enabled when the contact list is deleted or when a user is removed from the contact list. 'No' means that the automatic subscription / unsubscription is disabled.
Range	Defined in section 7.1.1

Table 2. AutoSubscribe

XML ELEMENT	CancelNote
Data type	String
Format	
Description	Short desciptive text for canceling of invitation.
Range	

Table 3. CancelNote

XML ELEMENT	ClientID
Data type	EMPTY
Format	
Description	Container element for possible client identification methods (url, msisdn etc..)
Range	

Table 4. ClientID

XML ELEMENT	ContactListID
Data type	EMPTY
Format	
Description	Container element for contactListID attribute.
Range	

Table 5. ContactListID

XML ELEMENT	ContactUser
Data type	EMPTY
Format	
Description	Defines a user in a contact list by user id and nickname.
Range	

Table 6. ContactUser

XML ELEMENT	ContactUserSpec
Data type	EMPTY
Format	
Description	Uniquely identifies a user in the contact list either by user id or nickname.
Range	

Table 7. ContactUserSpec

XML PCDATA	ContentID
Data type	EMPTY
Format	Defined in [CSP].
Description	Identifies a shared content by url attribute.
Range	

Table 8. ContentID

XML PCDATA	DateTime
Data type	Date and Time
Format	Specified in the forat attribute.
Description	Date and time.
Range	

Table 9. DateTime

XML ELEMENT	DeliveryTime
Data type	DateTime
Format	Specified in the format attribute
Description	Date and time of message delivery.
Range	

Table 10. DeliveryTime

XML ELEMENT	GroupID
Data type	EMPTY
Format	
Description	Container element for groupID attribute.
Range	

Table 11. GroupID

XML ELEMENT	JoinedUser
Data type	String
Format	
Description	Specifies a joined user of a group with screen name and UserID. The element contains the actual screen name of the user in the specified group, the userID attribute contains the UserID.
Range	

Table 12. JoinedUser

XML ELEMENT	MessageSpec
Data type	EMPTY
Format	
Description	Uniquely identifies a message by message id or url.
Range	

Table 13. MessageSpec

XML ELEMENT	PasswordDigest
Data type	Binary
Format	Defined in encoding attribute. BASE64 by default.
Description	The password digest used in login phase. See [SSP]
Range	

Table 14. PasswordDigest

XML ELEMENT	PresenceSubList
Data type	String
Format	As defined in the referred namespace.
Description	Presence attribute list with or without values. This element is used to give reference to the namespace (DTD) to be used under this specific tag.
Range	The namespace attribute points to a valid Wireless Village presence namespace.

Table 15. PresenceSublist

XML ELEMENT	Property
Data type	String
Format	
Description	Text containing a property information identified by the prop attribute and its value is contained in the element value.
Range	

Table 16. Property

XML ELEMENT	ReasonText
Data type	String
Format	
Description	Text describing a reason of an operation.
Range	

Table 17. ReasonText

XML ELEMENT	SearchTerm
Data type	EMPTY
Format	
Description	Container element for search terms. Contains an attribute type (attr) and a value which the search operation must be carried out.
Range	

Table 18. SearchTerm

XML ELEMENT	ScreenName
Data type	String
Format	
Description	Defines the screen name of a user in a group. Group is specified in the groupID attribute. Element content is case insensitive.
Range	

Table 19. ScreenName

XML ELEMENT	SecretToken
Data type	Binary
Format	Defined in encoding attribute. By default BASE64 encoded
Description	Secret token used in login phase.
Range	

Table 20. SecretToken

XML ELEMENT	StatusDescription
Data type	String
Format	
Description	Text containing a status description in reponse of a request.
Range	

Table 21. StatusDescription

XML ELEMENT	UPInfo
Data type	String
Format	
Description	Text containing a user profile information indetified by the attr attribute and its value is contained in the element value.
Range	

Table 22. UPInfo

XML ELEMENT	UserID
Data type	EMPTY
Format	
Description	Container element for userID attribute.
Range	

Table 23. UserID

XML ELEMENT	Version
Data type	EMPTY
Format	
Description	Specifies a version in the element vale. Format specified in the format attribute. By default format is DateTime defined in [CSP].
Range	

Table 24. Version

XML ELEMENT	WelcomeNote
Data type	String
Format	
Description	Welcome text of a group.
Range	

Table 25. WelcomeNote

7.4   Description of composed XML Elements

In this section will provide the descriptions of the composed XML elements that are neither assigned to an informational element nor are terminal elements.

XML Element	AuthRequestTuple
Description	This element contains the Authorizer (Authorizing-User-ID informational element), the Subscribers (List-of-Subscribing-User-ID informational element) and the AttributeList (Presence-Attribute-List informational element). It means who (Subscribers) want what information (AttributeList) about which user (Authorizer). One AuthorizationRequest primitive can contain more AuthRequestTuples. Different AuthRequestTuples represent different authorization requests.

XML Element	AuthResponseTuple
Description	This element contains the Authorizer (Authorizing-User-ID informational element) and the SubsciberResults (Subscribing-User-ID-List and Authorization-Result informational elements together). It means who responses to the request (Authorizer) and who he allows or not (SubscriberResult). One AuthorizationResponse primitive can contain more AuthRequestTuples. Different AuthResponseTuples represent different authorization responses.

XML Element	PresenceValue
Description	Contains a list of presence attributes along with its values (Presence-Value-List informational element) and an optional version information (Presence-Value-Version informational element)

XML Element	Recipient
Description	Defines the recipient of an instant message. The instant message must be delivered to the address defined in this field. Can contain an optinal RecipientDisplay element desciping what destination should be displayed to the recipient user (e.g. the group name where the message was originally sent).

XML Element	Requestor
Description	Represents the requestor of a request primitive.

XML Element	Sender
Description	Describes the sender of an instant message. Can contain the optional SenderDisplay element describing what source should be displayed to the recipient user.

XML Element	Session
Description	Session element represents transactions sended in the same sessions. Its sessionID attribute dentifies the session between the requestor and the provider server.

XML Element	SetupTransaction
Description	Contains a transaction outside of n existing session. Possible child elements are the transactions for session setup.

XML Element	SubscriberResult
Description	Defines the result of one authorization request. Contains the UserID who is allowed/forbidden to access the defined attribute and the granted attribute that defines if the access was allowed or not.

XML Element	Transaction
Description	Contains an SSP transaction primitive.

XML Element	User
Description	Represents a specific user using a specific terminal.

XML Element	VerUserID
Description	Represents a versioned user identifier (User-ID-List informational element). See description of Presence-Value-Version-List in [SSP].

XML Element	VerContactListID
Description	Represents a versioned contact list identifier (Contact-List-ID-List informational element). See description of Presence-Value-Version-List in [SSP].

XML Element	WV-SSP-Message
Description	Represents the SSP protocol message

8. XML Binding Examples

Examples of selected SSP transactions are presented in this section.

8.1 Login Transaction

```xml
<!-- Message from ServerA to ServerB -->
<WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2">
 <SetupTransaction mode="Request" transactionID="0">
 <SendSecretToken serviceID="wv@operator.hu" protocol="WV-SSP"
  procotolVersion="1.2">
   <SecretToken encoding="base64">

  R5R5FHJF47RY838289290050W0R989E0ER0R9E0392848GFJF8484JF84839294U745723934U
  7
   </SecretToken>
  </SendSecretToken>
 </SetupTransaction>
</WV-SSP-Message>


<!-- Message from ServerB to ServerA -->
<WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2">
 <SetupTransaction mode="Response" transactionID="0">
  <LoginRequest serviceID="wv@operator.hu" timeToLive="1314000">
   <PasswordDigest
   encoding="base64">FJS823HFUGF73HFBNE73BCJED34E60F5504W560001929394
E52</PasswordDigest>
  </LoginRequest>
 </SetupTransaction>
</WV-SSP-Message>


<!-- Message from ServerA to ServerB -->
<WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2">
 <SetupTransaction mode="Response" transactionID="0">
  <LoginResponse sessionID="47563@someone.za" timeToLive="1314000">
   <Status code="200"/>
  </LoginResponse>
 </SetupTransaction>
</WV-SSP-Message>
```

8.2 Status primitive with details

```xml
<WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2">
<Session sessionID="123456@pgsm.hu">
 <Transaction mode="Response" transactionID="420042">
  <Status code="400">
   <StatusDescription>Bad request</StatusDescription>
  </Status>
 </Transaction>
 <Transaction mode="Response" transactionID="420043">
  <Status code="200"/>
 </Transaction>
</Session>
</WV-SSP-Message>
```

8.3   Logout Transaction

```
<WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2">
<Session sessionID="123456@someone.za">
 <Transaction mode="Request" transactionID="5006">
  <LogoutRequest/>
 </Transaction>
</Session>
</WV-SSP-Message>

<WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2">
<Session sessionID="123456@operator.hu">
 <Transaction mode="Response" transactionID="5006">
  <Disconnect>
   <Status code="200"/>
  </Disconnect>
 </Transaction>
</Session>
</WV-SSP-Message>
```

8.4   KeepAlive transaction

```
<WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2">
<Session sessionID="123456@someone.za">
 <Transaction mode="Request" transactionID="5003">
  <KeepAliveRequest timeToLive="3600"/>
 </Transaction>
</Session>
</WV-SSP-Message>

<WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2">
<Session sessionID="123456@operator.hu">
 <Transaction mode="Response" transactionID="5003">
  <KeepAliveResponse timeToLive="3600">
   <Status code="200"/>
  </KeepAliveResponse>
 </Transaction>
</Session>
</WV-SSP-Message>
```

8.5   GetAvailableService transaction

```
<WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2">
<Session sessionID="123456@someone.za">
 <Transaction mode="Request" transactionID="5002">
  <GetServiceRequest/>
 </Transaction>
</Session>
</WV-SSP-Message>


<WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2">
<Session sessionID="123456@operator.hu">
 <Transaction mode="Response" transactionID="5002">
```

```
       <ServiceList>
        <Status code="200"/>
        <ServiceTree>
          <SRV_SAP/>
       <SRV_Common/>
       <SRV_Presence/>
       <SRV_IM/>
       <SRV_Group/>
       <ServiceTree>
       </ServiceList>
   </Transaction>
   </Session>
   </WV-SSP-Message>
```

8.6   ServiceIndication transaction

```
   <WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2">
   <Session sessionID="123456@operator.hu">
    <Transaction mode="Response" transactionID="5002">
     <ServiceList>
      <Status code="200"/>
      <ServiceTree>
        <SRV_SAP/>
      <SRV_Common/>
      <SRV_Presence/>
      <SRV_IM/>
      <SRV_Group/>
      <ServiceTree>
   </ServiceList>
   </Transaction>
   </Session>
   </WV-SSP-Message>
```

8.7   SetServiceAgreement transaction

```
   <WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2">
   <Session sessionID="123456@someone.za">
    <Transaction mode="Request" transactionID="113">
     <ServiceNegotiation timeToLive="3600">
      <ServiceTree>
        <SRV_SAP/>
      <SRV_Common/>
      <SRV_Presence/>
      <SRV_IM/>
      <SRV_Group/>
      <ServiceTree>
      </ServiceNegotiation>
    </Transaction>
   </Session>
   </WV-SSP-Message>


   <WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2">
   <Session sessionID="123456@operator.hu">
    <Transaction mode="Response" transactionID="113">
     <ServiceAgreement timeToLive="3600">
```

```
 <Status code="200"/>
 <ServiceTree>
   <SRV_SAP/>
 <SRV_Common/>
 <SRV_Presence/>
 <SRV_IM/>
 <SRV_Group/>
 <ServiceTree>
 </ServiceAgreement>
</Transaction>
</Session>
</WV-SSP-Message>
```

8.8   GetUserProfile transaction

```
<WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2">
<Session sessionID="123456@someone.za">
 <Transaction mode="Request" transactionID="113">
  <GetUserProfileRequest userID="wv:doboz.bela@operator.za">
   <MetaInfo clientOriginated="Yes">
    <Requestor>
      <User userID="wv:quador@operator.hu">
     <ClientID url="http://www.wireless-village.org/applications/MudServer"
  />
      </User>
    </Requestor>
   </MetaInfo>
  </GetUserProfileRequest>
 </Transaction>
</Session>
</WV-SSP-Message>
```


```
<WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2">
<Session sessionID="123456@operator.hu">
 <Transaction mode="Response" transactionID="113">
  <UserProfile>
   <Status code="200"/>
   <UserProfileValue userID="wv:doboz.bela@operator.za">
    <UPInfo attr="Services.Presence.Status">ON</UPInfo>
    <UPInfo attr="Services.Common.VHSE">pgsm.hu</UPInfo>
   </UserProfileValue>
  </UserProfile>
 </Transaction>
</Session>
</WV-SSP-Message>
```

8.9   UpdateUserProfile transaction

```
<WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2">
<Session sessionID="123456@someone.za">
 <Transaction mode="Request" transactionID="113">
  <UpdateUserProfileRequest userID="wv:doboz.bela@operator.za">
   <MetaInfo clientOriginated="Yes">
    <Requestor>
      <User userID="wv:quador@operator.hu">
```

```
     <ClientID url="http://www.wireless-village.org/applications/MudServer"
   />
       </User>
     </Requestor>
    </MetaInfo>
    <UserProfileValue>
     <UPInfo attr="Services.Presence.Status">ON</UPInfo>
     <UPInfo attr="Services.Common.VHSE">pgsm.hu</UPInfo>
    </UserProfileValue>
   </GetUserProfileRequest>
 </Transaction>
</Session>
</WV-SSP-Message>


<WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2">
<Session sessionID="123456@operator.hu">
 <Transaction mode="Response" transactionID="113">
  <Status code="200"/>
 </Transaction>
</Session>
</WV-SSP-Message>
```

Appendix A.   Static Conformance Requirements   (Normative)

The static conformance requirements for this specification is specified in [CSP SCR] and [SSP SCR].

Appendix B. Change History (Informative)

Type of Change	Date	Section	Description
Class 0	2003-01-17		The initial version of this document.