



WV-034 SSP - Transport Binding Version 1.1

Draft Version 10-01-2002

Open Mobile Alliance
OMA-WV-SSP_Syntax-V1_1-20021001-A

Continues the Technical Activities
Originated in the Wireless Village Initiative



This document is considered confidential and may not be disclosed in any manner to any non-member of the Open Mobile Alliance™, unless there has been prior explicit Board approval.

This document is a work in process and is not an approved Open Mobile Alliance™ specification. This document is subject to revision or removal without notice. No part of this document may be used to claim conformance or interoperability with the Open Mobile Alliance specifications.

© 2002, Open Mobile Alliance Ltd. All rights reserved.

Terms and conditions of use are available from the Open Mobile Alliance™ Web site at <http://www.openmobilealliance.org/copyright.html>.

You may use this document or any part of the document for internal or educational purposes only, provided you do not modify, edit or take out of context the information in this document in any manner. You may not use this document in any other manner without the prior written permission of the Open Mobile Alliance™. The Open Mobile Alliance authorises you to copy this document, provided that you retain all copyright and other proprietary notices contained in the original materials on any copies of the materials and that you comply strictly with these terms. This copyright permission does not constitute an endorsement of the products or services offered by you.

The Open Mobile Alliance™ assumes no responsibility for errors or omissions in this document. In no event shall the Open Mobile Alliance be liable for any special, indirect or consequential damages or any damages whatsoever arising out of or in connection with the use of this information.

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.html>.

Document History	
OMA-WV-SSP_Syntax-V1_1-20021001-A	Current

Contents

1. SCOPE.....	4
2. REFERENCES	5
2.1 NORMATIVE REFERENCES	5
2.2 INFORMATIVE REFERENCES	5
3. TERMINOLOGY AND CONVENTIONS	6
3.1 CONVENTIONS	6
3.2 DEFINITIONS.....	6
3.3 ABBREVIATIONS.....	6
4. INTRODUCTION.....	7
5. DOCUMENT TYPE DEFINITION.....	8
6. DATA TYPE ASSIGNMENTS FOR SSP XML ELEMENTS	21
7. DATA TYPES FOR XML ELEMENTS AND ATTRIBUTES	26
7.1 REDEFINED AND NEW DATA TYPES	26
7.1.1 Boolean.....	26
7.1.2 MSISDN	26
7.1.3 URL	26
7.2 DATA TYPES FOR XML ATTRIBUTES.....	26
7.3 DATA TYPES FOR XML TERMINAL ELEMENTS	35
7.4 DESCRIPTION OF COMPOSED XML ELEMENTS	39
8. XML BINDING EXAMPLES	43
8.1 LOGIN TRANSACTION.....	43
8.2 STATUS PRIMITIVE WITH DETAILS.....	43
8.3 LOGOUT TRANSACTION.....	44
8.4 KEEP ALIVE TRANSACTION.....	44
8.5 GETAVAILABLESERVICE TRANSACTION.....	44
8.6 SERVICEINDICATION TRANSACTION.....	45
8.7 SETSERVICEAGREEMENT TRANSACTION.....	45
8.8 GETUSERPROFILE TRANSACTION.....	46
8.9 UPDATEUSERPROFILE TRANSACTION.....	46
APPENDIX A. STATIC CONFORMANCE REQUIREMENTS (NORMATIVE)	48
APPENDIX B. CHANGE HISTORY (INFORMATIVE)	49

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://www.wapforum.org/](http://www.wapforum.org/) <to be replaced by an OMA ref when available>
- [RFC822] “Standard for the Format of ARPA Internet Text Messages”, August 1982. [URL:http://www.ietf.org/rfc/rfc822.txt?number=822](http://www.ietf.org/rfc/rfc822.txt?number=822)
- [RFC2119] “Keywords for using RFCs to Indicate Requirements levels”, Bradner, S. [URL:http://www.ietf.org/rfc/rfc2119.txt?number=2119](http://www.ietf.org/rfc/rfc2119.txt?number=2119)
- [RFC2778] “A Model for Presence and Instant Messaging”, February 2000. [URL:http://www.ietf.org/rfc/rfc2778.txt?number=2778](http://www.ietf.org/rfc/rfc2778.txt?number=2778)
- [XML] “Extensible Markup Language 1.0 (Second Edition)”, W3C recommendation, October-2000.
- [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](http://www.ietf.org/rfc/rfc2046.txt?number=2046)
- [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](http://www.ietf.org/rfc/rfc2045.txt?number=2045)
- [RFC2396] Uniform Resource Identifiers (URI): Generic Syntax. [URL:http://www.ietf.org/rfc/rfc2396.txt?number=2396](http://www.ietf.org/rfc/rfc2396.txt?number=2396)
- [FeaFun] "WV-021 Features and Functions Version 1.1". October 2002. [URL:http://www.openmobilealliance.org/member/technicalPlenary/imps/docs/OMA-WV-Features_Functions-V1_1-20021001-A](http://www.openmobilealliance.org/member/technicalPlenary/imps/docs/OMA-WV-Features_Functions-V1_1-20021001-A)
- [CSP] "WV-022 Client-Server Protocol Session and Transactions Version 1.1". October 2002. [URL:http://www.openmobilealliance.org/member/technicalPlenary/imps/docs/OMA-WV-CSP-V1_1-20021001-A](http://www.openmobilealliance.org/member/technicalPlenary/imps/docs/OMA-WV-CSP-V1_1-20021001-A)
- [CSP DataType] "WV-025 Client-Server Protocol Data Types ". October 2002. [URL:http://www.openmobilealliance.org/member/technicalPlenary/imps/docs/OMA-WV-CSP_DataTypes-V1_1-20021001-A](http://www.openmobilealliance.org/member/technicalPlenary/imps/docs/OMA-WV-CSP_DataTypes-V1_1-20021001-A)
- [PA] "WV-029 Presence Attributes Version 1.1". October 2002. [URL:http://www.openmobilealliance.org/member/technicalPlenary/imps/docs/OMA-WV-PA-V1_1-20021001-A](http://www.openmobilealliance.org/member/technicalPlenary/imps/docs/OMA-WV-PA-V1_1-20021001-A)

2.2 Informative References

- [WAPARCH] “WAP Architecture”. Open Mobile Alliance™. WAP-210-WAPArch. [URL:http://www.wapforum.org/](http://www.wapforum.org/)

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 document 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) V1.1 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 1.1//EN"
    "http://www.openmobilealliance.org/DTD/WV-SSP.DTD"
    [<?oma-wv-ssp-ver supported-versions="1.1"?>]>
  <WV-SSP>
  . . .
  </WV-SSP>
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)>
  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 "1.1">
<!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
  timeToLive CDATA #IMPLIED>
<!ELEMENT PasswordDigest (#PCDATA)>
<!ATTLIST PasswordDigest
  encoding CDATA "base64">

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

```



```

<!-- 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 |
  InviteUserRequest | InviteUserResponse | CancelInviteUserRequest |
  CreateContactListRequest | DeleteContactListRequest |
  GetContactListRequest | GetContactListResponse | GetListMemberRequest |
  AddListMemberRequest | RemoveListMemberRequest | ContactListMemberResponse
  | GetListPropsRequest | SetListPropsRequest | ContactListPropsResponse |
  CreateAttrListRequest | DeleteAttrListRequest | GetAttrListRequest |
  GetAttrListResponse | AuthorizationRequest | AuthorizationResponse |
  CancelAuthRequest | SubscribeRequest | UnsubscribeRequest |
  GetWatcherListRequest | GetWatcherListResponse | PresenceNotification |
  GetPresenceRequest | GetPresenceResponse | UpdatePresenceRequest |
  SendMessageRequest | SendMessageResponse | ForwardMessageRequest |
  NewMessage | MessageDelivered | MessageNotification | GetMessageRequest |
  SetMessageDeliveryMethod | GetMessageListRequest | GetMessageListResponse
  | RejectMessageRequest | DeliveryStatusRequest | DeliveryStatusReport |
  BlockUserRequest | GetBlockedRequest | GetBlockedResponse |
  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_Presence?, SRV_SharedContent?, SRV_IM?,
    SRV_Group?)>
<!ELEMENT SRV_Presence EMPTY>
<!ELEMENT SRV_SharedContent EMPTY>
<!ELEMENT SRV_IM EMPTY>
<!ELEMENT SRV_Group 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?)>
<!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_IM (SRV_SendMessage?, SRV_PushMessage?,
  SRV_MessageNotification?, SRV_GetMessage?, SRV_SetMessageDeliveryMethod?,
  SRV_GetMessageList?, SRV_RejectMessage?, SRV_DeliveryReport?,
  SRV_Blocking?, SRV_GroupHistory?)>
<!ELEMENT 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) #REQUIRED
    validity        CDATA            #IMPLIED>
<!ELEMENT Inviting (User | ScreenName)>
<!ELEMENT Invited (User | ScreenName | ContactListID)>
<!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) #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>

<!-- VerifyUserIDRequest -->
<!ELEMENT VerifyUserIDRequest (MetaInfo, VerifyUserID+)>
<!ELEMENT VerifyUserID (DateTime?)>
<!ATTLIST VerifyUserID
  userID CDATA      #REQUIRED>
<!ELEMENT DateTime (#PCDATA)>
<!ATTLIST DateTime
  format      CDATA      "iso8601">

<!-- VerifyUserIDResponse -->
<!ELEMENT VerifyUserIDResponse (Status, VerifyUserID*)>

<!-- ***** -->
<!-- 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?)>

<!-- 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)>
<!ATTLIST SubscriberResult
  grantedCDATA #REQUIRED>

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

```

```

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

<!-- 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?)>
<!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		granted (of SubscriberResult)
Authorizing-User-ID	Authorizer	
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	

Default-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-Subscribing-User-ID-and-Presence-Attribute-List	AuthRequestTuple/Subscribers and AuthRequestTuple/AttributeList	

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	
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

Verify-User-ID-List	VerifyUserID+	
Welcome -Text	WelcomeNote	

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 [WV-DataTypes].

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 [WV-SSP] Section 12.2.8.

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 [WV-SSP] Section 12.2.14.

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 Section 9. of the [WV-CSP].

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 [WV-SSP] Section 3.3.3. 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 [WV-SSP] and [WV-CSP] Section 7.1.2.1.

XML Attribute	deliveryReport
Data type	Boolean
Description	Specifies if a delivery report is requested for the send 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 [WV-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 [WV-DataTypes].

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 [WV-SSP] Section 12.2.14.

XML Attribute	groupID
Data type	String
Description	Corresponds to the Group-ID defined in [WV-SSP] Section 3.3.4. Case insensitive.

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

XML Attribute	inviteType
Data type	Enum
Range	"GR" "IM" "PR" "SC"
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.

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 [WV-SSP] Section 13.1.3.

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 protocol used. Must be "WV-SSP".

XML Attribute	protocolVersion
Data type	String
Range	"1.1"
Description	Defines the version of the protocol used. Must be "1.1".

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

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

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

XML Attribute	searchLimit
Data type	Integer
Description	See in [WV-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 [WV-SSP] Section 3.3.7.
Description	See [WV-SSP] Section 3.3.7. 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 [WV-SSP] Section 3.3.2. 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 [WV-DataTypes]. If an XML terminal element definition is found in both this document and in [WV-DataTypes] 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 the Section 7.1 of the [WV-SSP] document.

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

Table 1. Authorizer

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

Table 2. CancelNote

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

Table 3. ClientID

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

Table 4. ContactListID

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

Table 5. 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 6. ContactUserSpec

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

Table 7. ContentID

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

Table 8. DateTime

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

Table 9. DeliveryTime

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

Table 10. 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 11. JoinedUser

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

Table 12. MessageSpec

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

Table 13. 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 14. 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 15. Property

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

Table 16. 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 17. 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 18. 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 19. SecretToken

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

Table 20. StatusDescription

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

Table 21. UPInfo

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

Table 22. 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 [WV-CSP].
Range	

Table 23. Version

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

Table 24. 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	<p>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).</p> <p>One AuthorizationRequest primitive can contain more AuthRequestTuples. Different AuthRequestTuples represent different authorization requests.</p>

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

XML Element	PresenceValue
Description	<p>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)</p>

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

XML Element	Requestor
Description	<p>Represents the requestor of a request primitive.</p>

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

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 [WV-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 [WV-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

```
<!-- Message from ServerA to ServerB -->
<WV-SSP-Message xmlns="http://www.wireless-village.org/SSP1.1">
  <SetupTransaction mode="Request" transactionID="0">
    <SendSecretToken serviceID="wv@operator.hu" protocol="WV-SSP"
      protocolVersion="1.1">
      <SecretToken encoding="base64">

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

```
<!-- Message from ServerB to ServerA -->
<WV-SSP-Message xmlns="http://www.wireless-village.org/SSP1.1">
  <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.wireless-village.org/SSP1.1">
  <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

```
<WV-SSP-Message xmlns="http://www.wireless-village.org/SSP1.1">
  <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.wireless-village.org/SSP1.1">
<Session sessionID="123456@someone.za">
  <Transaction mode="Request" transactionID="5006">
    <LogoutRequest/>
  </Transaction>
</Session>
</WV-SSP-Message>
```

```
<WV-SSP-Message xmlns="http://www.wireless-village.org/SSP1.1">
<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.wireless-village.org/SSP1.1">
<Session sessionID="123456@someone.za">
  <Transaction mode="Request" transactionID="5003">
    <KeepAliveRequest timeToLive="3600"/>
  </Transaction>
</Session>
</WV-SSP-Message>
```

```
<WV-SSP-Message xmlns="http://www.wireless-village.org/SSP1.1">
<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.wireless-village.org/SSP1.1">
<Session sessionID="123456@someone.za">
  <Transaction mode="Request" transactionID="5002">
    <GetServiceRequest/>
  </Transaction>
</Session>
</WV-SSP-Message>
```

```
<WV-SSP-Message xmlns="http://www.wireless-village.org/SSP1.1">
<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.wireless-village.org/SSP1.1">
<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.wireless-village.org/SSP1.1">
<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.wireless-village.org/SSP1.1">
<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.wireless-village.org/SSP1.1">
<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.wireless-village.org/SSP1.1">
<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.wireless-village.org/SSP1.1">
<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.wireless-village.org/SSP1.1">
<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	2002-10-25		The initial version of this document.