



Client-Server Protocol
SMS Binding
Version 1.1

WV Internal Tracking Number: WV-026

Notice

Copyright © 2001-2002 **Ericsson, Motorola and Nokia**. All Rights Reserved.

Implementation of all or part of any Specification may require licenses under third party intellectual property rights, including without limitation, patent rights (such a third party may or may not be a Supporter). The Sponsors of the Specification are not responsible and shall not be held responsible in any manner for identifying or failing to identify any or all such third party intellectual property rights.

THIS DOCUMENT AND THE INFORMATION CONTAINED HEREIN ARE PROVIDED ON AN "AS IS" BASIS WITHOUT WARRANTY OF ANY KIND AND ERICSSON, MOTOROLA and NOKIA DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL ERICSSON, MOTOROLA or NOKIA BE LIABLE TO ANY PARTY FOR ANY LOSS OF PROFITS, LOSS OF BUSINESS, LOSS OF USE OF DATA, INTERRUPTION OF BUSINESS, OR FOR DIRECT, INDIRECT, SPECIAL OR EXEMPLARY, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES OF NY KIND IN CONNECTION WITH THIS DOCUMENT OR THE INFORMATION CONTAINED HEREIN, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE. The above notice and this paragraph must be included on all copies of this document that are made.

Intellectual Property Rights have been asserted or conveyed in some manner toward these Wireless Village specifications. The Wireless Village initiatives' intellectual property rights guidelines are defined in Section 5.1 of the Wireless Village Specification Supporter Agreement. The Wireless Village initiative takes no position regarding the validity or scope of any intellectual property right or other rights that might be claimed to pertain to the implementation or use of the technology, or the extent to which any license under such rights might or might not be available. A public listing of all claims against the Wireless Village specifications, as well as an excerpt of Section 5.1 of the Wireless Village Specification Supporter Agreement, can be found at:

<http://www.wireless-village.org/ipr.html>

Table of contents

1.	Revision History	1
2.	References.....	2
3.	Introduction	3
4.	SMS binding.....	4
4.1	Syntax	4
4.2	Encoding of Short Messages	6
4.3	Handling of Multiple Short Messages within Transaction	7
5.	Availability of transactions	8
6.	SMS-specific encoding	11
6.1	Information elements	11
6.2	Service tree elements	13
6.3	Presence attributes.....	15
6.4	Presence values	16
6.5	Group properties	16
6.6	Contact list properties	17
6.7	Search elements.....	17
7.	SMS Binding Examples	19
7.1	Status primitive	19
7.2	Login transaction	19
7.3	Logout transaction	20
7.4	Server initiated logout transaction	20
7.5	Keep Alive transaction.....	20
7.6	Get service provider info transaction.....	20
7.7	Service request transaction.....	20
7.8	Search transaction.....	21
7.9	Stop search transactions	22
7.10	Invitation transactions	22
7.11	Canceling invitation transactions.....	23
7.12	Get list of contact list IDs transaction.....	23
7.13	Create contact list transaction.....	23
7.14	Delete contact list transaction	23
7.15	Retrieve a contact list transaction	24
7.16	Add users to a contact list transaction	24
7.17	Remove users from a contact list.....	24
7.18	Modify properties of contact list transaction.....	25
7.19	Create attribute list transaction.....	25
7.20	Delete attribute list transaction	25
7.21	Get attribute list(s) transaction	25
7.22	Subscribe/unsubscribe presence transaction	26
7.23	Get watcher list transaction	27
7.24	Get presence transaction	27
7.25	Reactive presence authorization transactions	27
7.26	Update presence transaction	28
7.27	Send message transaction.....	28
7.28	Pushing a message from the server transaction.....	28
7.29	Get message list transaction	28
7.30	Retrieving a message from the server transaction	29
7.31	Delivery status report transaction.....	29
7.32	Get blocked user list transaction	29
7.33	Block entity transaction.....	30
7.34	Create group transaction	30

7.35	Delete group transaction	30
7.36	Join Group transaction	31
7.37	User initiated leave group transaction.....	31
7.38	Server initiated leave group transaction.....	31
7.39	Get group members' list transaction	31
7.40	Add group member(s) transaction.....	32
7.41	Remove group member(s) transaction.....	32
7.42	Member access rights transaction.....	32
7.43	Modify group properties transactions	32
7.44	Rejected list transactions.....	33
7.45	Subscribe group change notification transaction	33
7.46	Example for multiple transactions	34
8.	Static Conformance Requirement for CSP SMS Binding:	35

1. REVISION HISTORY

Date	Issue	Description	Author
February 13 th	TBD	Initial release	WV TechComm
July 31, 2002	V1.1	Version 1.1	WV TechComm

2. REFERENCES

- [WV-CSP] Wireless Village Client-Server Protocol Session and Transactions specification
- [WV-CSP-TRANS] Wireless Village – Client-Server protocol, Transport Binding
- [WV-PA] Wireless Village – Presence Attributes specification

3. INTRODUCTION

This document describes how the SMS can be utilized to provide WV functionality.

The message flows are defined in [WV-CSP].

The enumerated values are as specified in [WV-CSP].

4. SMS BINDING

4.1 SYNTAX

General SMS message format consists of preamble and parameters as follows:

WVaaBBcccDD <parameters> [&] where:

- **WV** indicates that this is a Wireless Village message. It is case sensitive.
- **aa** is the version number of the IM specification. Major version and minor version numbers without the dot in the middle, both are single digits from 0 to 9. This specification uses version number 1.1 (11 in messages).
- **BB** is the message type, identified by each message table. It is not case sensitive.
- The **ccc** is the Transaction-ID in range 0-999 without preceding zero.
- The **DD** identifies multiple short messages within single transaction; it is a two-letter identifier within the range of a-z. The first letter indicates the number (position) of the message; the second letter indicates the last part. (This gives the possibility to split a message into 26 SMSes, which limits the size of the message to $26 \times 160 = 4160$ characters. Note that it includes every additional information elements such as tags, commas and so on, so not the whole capacity is available to the user.) It is not case sensitive.
- The ampersand (&) character separates multiple WV messages in one short message. The separating ampersand does not follow the last WV message in the short message.

The general format of parameters shall be:

<name>=<value>

After the *aaBBcccDD* preamble, there is one space character. Each parameter is separated from other parameters by single space character.

If value contains spaces (), quotes ("), commas (,), parentheses (()), equal (=) or ampersand (&) characters, it shall be wrapped with quotes (").

If value contains quotes ("), all quote characters shall be doubled.
For example if the value is:

John "Johnnie" Smith

then it shall be encoded as:

"John ""Johnnie"" Smith"

If the value is only one single quote, the encoding would be four quotes: """"

A parameter may also contain a list of values. In this case, the syntax shall be (example with three values):

<name>=(<value1>,<value2>,<value3>)

A parameter may also contain a group of values. These groups may be also nested.
Examples:

one value having three values:

```
<name>=(<value1>,<value2>,<value3>,<value4>))
```

nested values:

```
<name>=(<value1>,<value2>,<value3>,<value4>)))
```

Definitions shall be case-sensitive.

4.2 ENCODING OF SHORT MESSAGES

The encoding of SMS messages has two alternatives: 7-bit encoding using the SMS default character set [TS 23.038] and 8-bit encoding using UTF-8 character encoding. With 8-bit encoding, the minimum supported set of characters is ISO-8859-1 (Latin 1).

Each WV short message may be either textual or contain a User Data Header (UDH) with the TP-UDHI value set to 1. In case of UDH, the detection of the WV-primitives is based on a registered WV application port identified in the headers. In case of textual short message, the detection of WV-primitives is based on detection of the message preamble. The User Data Header contains a 16-bit application port number identifying the source port and destination port. The structure of the short message with UDH is defined as follows [TS 23.040]:

The destination port number is assigned in the CSP transport-binding document [WV-CSP-TRANS]. When the session is started with a login primitive containing the UDH, the WV server and client must continue the session using the messages encoded with UDH throughout the session. Similarly, when the session is started with a login primitive without UDH, the WV server and client must continue the session without UDH.

UDL	UDHL	IEI	IEIDL	IED	UD
-----	------	-----	-------	-----	----

UDL Length of the message

UDHL Length of User Data Header (7 for WV message)

IEID Information Element Identifier (05_{hex} = 16-bit application port)

IEDL Length of IED (4 = four octets for ports)

IED Port numbers (octet 1,2 = destination port, octet 3,4=originator port)

UD User Data (WV primitive)

The source and destination port have same port numbers, assigned in the CSP transport-binding document [WV-CSP-TRANS].

4.3 HANDLING OF MULTIPLE SHORT MESSAGES WITHIN TRANSACTION

A transaction over short message bearer may be split to more than one short message when parameters exceed 160 characters. To accomplish this, there are two basic techniques available:

1. Use of concatenated short messages (SM-TP level concatenation).
2. Use concatenation identifier (**DD**) identifier in a short message text.

Alternative 1) is based on SMS technology, which is optionally supported in the SMSC and terminal product. It does not require support in text level.

Alternative 2) needs to be used when SMS concatenation is not available. The rules are:

- If the short message does not need text-level concatenation (single SMS or SM-TP level concatenation), the **DD** identifier is not present.
- If the short message needs text-level concatenation, the **DD** identifier is present (starting from 'a') and the first letter identifies the current SMS within concatenation sequence and the second letter identifies the last SMS within the sequence.
- While transferring the multiple short messages concatenated in a text level, the transaction identifier will be the same in all of the short messages.
- The request or response is incomplete until all short messages within the same transaction are received.

The text-level concatenation identifier does not require that the messages will be received in order. An example of concatenation could be the terminal sending a message:

```
WV11NM23ac MC="This is a very  
WV11NM23cc very long textual content..."  
WV11NM23bc very long message, and it has very
```

And the server responding:

```
WV11ST23 ST=200
```

A single short message may also contain messages from multiple transactions as defined according to the [WV-TRANS] document. In such cases, messages are separated by an ampersand (&) according to the syntax. The preamble of each WV message must fit as a whole within the SMS message.

5. AVAILABILITY OF TRANSACTIONS

The SMS protocol itself provides some features and lacks others, which simplifies or disables certain transactions.

All instant messages that contain plain text message content and fit into the limits specified in 4.1 Syntax without truncation should be pushed to the terminal.

The following table describes the availability of each primitive defined in [WV-CSP]:

Transaction	Support	Code
AddGroupMembersRequest	Full	AM
BlockEntityRequest	Full	BE
CancelAuthRequest	Full	CR
CancelInviteRequest	Full	CI
CancelInviteUserRequest	Full	CU
ClientCapabilityRequest	N/A	
ClientCapabilityResponse	N/A	
CreateAttributeListRequest	Full	CA
CreateGroupRequest	Full	CG
CreateListRequest	Full	CL
DeleteAttributeListRequest	Full	DA
DeleteGroupRequest	Full	DG
DeleteListRequest	Full	DL
DeliveryReportRequest	Full	DR
Disconnect	Full	DI
GetAttributeListRequest	Full	GA
GetAttributeListResponse	Full	AG
GetBlockedListRequest	Full	GB
GetBlockedListResponse	Full	BG
GetGroupMembersRequest	Full	GM
GetGroupMembersResponse	Full	MG
GetGroupPropsRequest	Full	GR
GetGroupPropsResponse	Full	RG
GetListRequest	Full	GL
GetListResponse	Full	LG
GetMessageListRequest	Full	MR
GetMessageListResponse	Simplified	RM
GetMessageRequest	Full	GX
GetMessageResponse	Simplified	MX
GetPresenceRequest	Full	GP
GetPresenceResponse	Full	PG
GetSPInfoRequest	Full	GS
GetSPInfoResponse	Simplified	SG
GetWatcherListRequest	Full	GW
GetWatcherListResponse	Full	WG
GroupChangeNotice	Full	GG
ForwardMessageRequest	N/A	
InviteRequest	Full	IR
InviteResponse	Full	RI
InviteUserRequest	Full	IU

InviteUserResponse	Full	UI
JoinGroupRequest	Full	JG
JoinGroupResponse	Full	GJ
KeepAliveRequest	Full	KA
KeepAliveResponse	Full	AK
LeaveGroupRequest	Full	LU
LeaveGroupResponse	Full	UL
ListManageRequest	Full	LM
ListManageResponse	Full	ML
LoginRequest	Simplified	LR
LoginResponse	Simplified	RL
LogoutRequest	Full	OR
MemberAccessRequest	Full	ME
MessageDelivered	Full	MD
MessageNotification	N/A	
NewMessage	Simplified	NM
PollingRequest	N/A	
PresenceAuthRequest	Full	PR
PresenceAuthUser	Full	RP
PresenceNotificationRequest	Full	PN
RemoveGroupMembersRequest	Full	RM
RejectListRequest	Full	RE
RejectListResponse	Full	ER
RejectMessageRequest	N/A	
SearchRequest	Full	SR
SearchResponse	Full	RS
SendMessageRequest	Simplified	SM
SendMessageResponse	Full	MS
ServiceRequest	Simplified	SQ
ServiceResponse	Simplified	QS
SetDeliveryMethodRequest	N/A	
SetGroupPropsRequest	Full	SP
Status	Full	ST
StopSearchRequest	Full	SS
SubscribeGroupNoticeRequest	Full	SU
SubscribeGroupNoticeResponse	Full	US
SubscribePresenceRequest	Full	SB
UnsubscribePresenceRequest	Full	PS
UpdatePresence	Full	UP

The codes are not case sensitive.

6. SMS-SPECIFIC ENCODING

6.1 INFORMATION ELEMENTS

Element	Code
Acceptance	AC
Add-Users-List	AU
Add-Nick-List	AN
Administrator	AD
Attribute-Association-Contact-List	AG
Attribute-Association-User-List	AL
All-Functions	AF
All-Functions-Request	AR
Blocked-Group-List	BG
Blocked-User-List	BL
Blocked-List-Inuse	BU
Client-ID	CI
Code	RC
CompletionFlag	CF
Contact-List-ID	CL
Contact-List-Props	CP
DateTime	DT
Default-Association-List	DA
Default-CList-ID	DC
Default-List	DL
Delivery-Report-Request	DE
Description	RT
Detailed-Result-Group	DG
Detailed-Result-Screenname	DS
Detailed-Result-User	DU
Delivery-Time	DX
Granted-Group-List	GG
Granted-User-List	GL
Granted-List-Inuse	GU
Group-ID	GI
Group-Props	GP
Invite-ID	II
Invite-Reason	IR
Invite-Response	IX
Invite-Type	IT
JoinGroup	JG
Joined-Request	JR
Joined-Users-List	JU
Keep-Alive-Time	KA
Left-Users-List	LU
Message-Content	MC
Message-Count	MN
Message-ID	MI
Moderator	MO
Name	NA

Not-Available-Functions	NF
Own-Props	OP
Own-Screen-Name	ON
Password-String	PW
Presence	PR
PresenceSubList	PS
Recalled-Content	RC
Recall-Reason	RR
Remove-Nick-List	RN
Remove-Users-List	RU
Requested-Functions	RF
Result (Status code and description)	ST
Screen-Name	SN
Search-Findings	SF
Search-ID	SD
Search-Index	SX
Search-Limit	SL
Search-Pair-List	SP
Search-Results	SR
Session-Cookie	SC
Session-ID	SI
SubscribeNotification	SA
Subscribe-Type	SU
Subscription-State	SS
Text	TX
Time-To-Live	TL
Unblock-Group-List	UB
Unblock-User-List	UU
Ungrant-Group-List	UG
Ungrant-User-List	UL
Update-Value-List	UV
URL	UR
User (ordinary access)	US
User-ID	UI
User-Nick-List	UN
User-Prop-List	UP
Validity	VA
Welcome-Text	WT

The codes are not case sensitive.

6.2 SERVICE TREE ELEMENTS

To shorten messages, the elements of the service tree have been encoded in shorter codes. The tree itself is the same as is specified in [WV-CSP]. There is one difference: the elements marked unsupported in the table below are not part of the SMS-specific service tree.

Name	Support	Code
ADDGM	Yes	AG
AttListFunc	Yes	AF
BLENT	Yes	BL
CAAUT	Yes	CA
CAINV	Yes	CI
CALI	Yes	CL
CCLI	Yes	CC
ContListFunc	Yes	FC
CREAG	Yes	CG
DALI	Yes	DA
DCLI	Yes	DC
DELGR	Yes	DG
FundamentalFeat	Yes	FF
FWMSG	No	
GALS	Yes	GA
GCLI	Yes	GC
GETGM	Yes	GG
GETGP	Yes	GP
GETLM	Yes	GL
GETM	Yes	GM
GETPR	Yes	GP
GETSPI	Yes	GS
GETWL	Yes	GW
GLBLU	Yes	GB
GRCHN	Yes	GN
GroupAuthFunc	Yes	GF
GroupFeat	Yes	GE
GroupMgmtFunc	Yes	GT
GroupUseFunc	Yes	GU
IMAuthFunc	Yes	IA
IMFeat	Yes	IF
IMReceiveFunc	Yes	IR
IMSendFunc	Yes	IS
INVIT	Yes	IV
InviteFunc	Yes	IN
MBRAC	Yes	MA
MCLS	Yes	MC
MDELIV	Yes	MD
NEWM	Yes	NM
NOTIF	No	
PresenceAuthFunc	Yes	PA
PresenceDeliverFunc	Yes	PD
PresenceFeat	Yes	PF
REACT	Yes	RA

REJCM	No	
REJEC	Yes	RE
RMVGM	Yes	RG
SearchFunc	Yes	SF
ServiceFunc	Yes	SU
SETD	No	
SETGP	Yes	SG
SRCH	Yes	SR
STSRC	Yes	ST
SUBGCN	Yes	SU
UPDPR	Yes	UP
WVCSPFeat	Yes	WV

The codes are not case sensitive.

6.3 PRESENCE ATTRIBUTES

Extension attributes are not supported, only presence attributes described in the [WV-PA].

Only plain text-based presence attributes are supported.

The qualifier is not present.

To shorten messages, the presence attributes have been encoded in shorter codes.

Name	Support	Code
Accuracy (GeoLocation)	Full	AL
Accuracy (Address)	Full	AA
Alias	Full	AI
Altitude	Full	AT
Building	Full	BU
City	Full	CI
ClientProducer	Full	CP
ClientType	Full	CT
ClientVersion	Full	CV
CommCap	Full	CC
ContactInfo	N/A	
Country	Full	CO
Crossing1	Full	C1
Crossing2	Full	C2
Cname	Full	CN
Cpriority	Full	CR
Cstatus	Full	CS
DevManufacturer	Full	DM
Extended Presence Info	N/A	
FreeTextLocation	Full	FT
Language	Full	LN
Latitude	Full	LA
TimeZone	Full	LT
Longitude	Full	LO
Model	Full	MO
NamedArea	Full	NA
OnlineStatus	Full	OS
PLMN	Full	PM
PreferredContacts	Full	PC
PreferredLanguage	Full	PL
Registration	Full	RG
StatusContent	N/A	
StatusMood	Full	SM
StatusText	Full	ST
Street	Full	SR
UserAvail	Full	UA

The codes are not case sensitive.

6.4 PRESENCE VALUES

To shorten messages, the enumerated presence values have been encoded in short codes.

Name	Code
ANGRY	AG
ANXIOUS	AX
ASHAMED	AS
AUDIO_CALL	AU
AVAILABLE	AV
BORED	BO
CALL	CA
CLI	CL
CLOSED	CS
COMPUTER	CO
DISCREET	DI
EMAIL	EM
EXCITED	EX
HAPPY	HA
IM	IM
IM_OFFLINE	OF
IM_ONLINE	ON
IN_LOVE	IL
INVINCIBLE	IN
JEALOUS	JE
MMS	MS
MOBILE_PHONE	MP
NOT_AVAILABLE	NA
OPEN	OP
OTHER	OT
PDA	PD
SAD	SA
SLEEPY	SL
SMS	SM
VIDEO_CALL	VC
VIDEO_STREAM	VS

The codes are not case sensitive.

6.5 GROUP PROPERTIES

To shorten messages, the group properties have been encoded in shorter codes.

Name	Code
AccessType	AT
ActiveUsers	AU
IsMember	IM
MaxActiveUsers	MU
Name	NM
PrivateMessaging	PM

PrivilegeLevel	PL
Searchable	SE
Topic	TO
Type	TY
WelcomeNote	WN
AutoJoin	AJ

The codes are not case sensitive.

6.6 CONTACT LIST PROPERTIES

To shorten messages, the contact list properties have been encoded in shorter codes.

Name	Code
DisplayName	DN
Default	DE

The codes are not case sensitive.

6.7 SEARCH ELEMENTS

To shorten messages, the enumerated search elements have been encoded in shorter codes.

Name	Code
GROUP_ID	GI
GROUP_NAME	GN
GROUP_TOPIC	GT
GROUP_USER_ID_JOINED	GJ
USER_ALIAS	UA
USER_EMAIL_ADDRESS	UE
USER_FIRST_NAME	UF
USER_ID	UI
USER_LAST_NAME	UL
USER_MOBILE_NUMBER	UM
USER_ONLINE_STATUS	UO
GROUP_USER_ID_OWNER	GO

The codes are not case sensitive.

7. SMS BINDING EXAMPLES

7.1 STATUS PRIMITIVE

With a single status code:

```
ST=<StatusCode>
```

```
WV11ST761 SI=im.user.com#48815@server.com ST=200
```

With a status code and description:

```
ST=(<StatusCode>,<Description>)
```

```
WV11ST761 SI=im.user.com#48815@server.com ST=(200,"Successfully completed.")
```

With multiple status codes and descriptions:

```
ST=(<StatusCode>,<Description>)
```

```
DG=(<StatusCode>,<Description>,<GroupID n>)
```

```
DS=(<StatusCode>,<Description>,<ScreenName n>,<GroupID n>)
```

```
DU=(<StatusCode>,<Description>,<UserID n>)
```

```
WV11ST761 SI=im.user.com#48815@server.com ST=(201,"Partially completed.")
DU=(531,"Unknown user.",wv:bad_user1@im.com,wv:bad_user1@im.com)
DU=(532,"Blocked.",wv:bad_user3@im.com,wv:bad_user4@im.com)
```

7.2 LOGIN TRANSACTION

This transaction is simplified, because there is no need to encrypt the password, since the bearer already provides the encryption.

7.2.1 LoginRequest primitive

Only the User-ID, the Client-ID, the password, the Session-Cookie, and the Time-To-Live elements are present in the primitive.

```
WV11LR761 UI=wv:john@smith.com CI=+1234567890 PW=this1is2my3pass
SC=im.user.com#20011224#328746293 TL=600
```

7.2.2 LoginResponse primitive

Only the Result, the Session-ID and the KeepAliveTime are present in the primitive. All other information elements are omitted.

```
WV11RL761 ST=(200,"Successfully completed.")
SI=im.user.com#48815@server.com
KA=300
```

7.3 LOGOUT TRANSACTION

7.3.1 LogoutRequest primitive

```
WV11OR761 SI=im.user.com#48815@server.com
```

7.3.2 Disconnect primitive

```
WV11DI761 SI=im.user.com#48815@server.com ST=200
```

7.4 SERVER INITIATED LOGOUT TRANSACTION

7.4.1 Disconnect primitive

```
WV11DI761 SI=im.user.com#48815@server.com ST=(601,"Updating
server software.
All services offline for 3 hours.")
```

7.5 KEEP ALIVE TRANSACTION

7.5.1 KeepAliveRequest primitive

```
WV11KA761 SI=im.user.com#48815@server.com TL=600
```

7.5.2 KeepAliveResponse primitive

```
WV11AK761 SI=im.user.com#48815@server.com ST=(200,"Successfully
completed.")KA=600
```

7.6 GET SERVICE PROVIDER INFO TRANSACTION

7.6.1 GetSPInfoRequest primitive

```
WV11GS761 SI=im.user.com#48815@server.com
```

7.6.2 GetSPInfoResponse primitive

MMS messages are not allowed, thus the Logo element is not present in the primitive.

```
WV11SG761 SI=im.user.com#48815@server.com NA="Wireless Village"
TX="This is Wireless Village's IMPS test service"
UR="http://www.wireless-village.org"
```

7.7 SERVICE REQUEST TRANSACTION

The following example illustrates service negotiation with the following parameters:

Client requests:

- All Fundamental Features,
- All IM Features,

- All Presence Features,
- None of the Group Features.

Server does not agree to provide:

- Fundamental features,
- GetWatcherList transaction,
- IM Authorization Functions.

7.7.1 ServiceRequest primitive

Client requests all functionality.

```
WV11SQ761 SI=im.user.com#48815@server.com CI=+1234567890
RF=(WV,(FF,IF,PF)) AR=F
```

7.7.2 ServiceResponse primitive

```
WV11QS761 SI=im.user.com#48815@server.com CI=+1234567890
NF=(WV,(FF,(PF,(PA,GW))),(IF,IA)))
```

7.8 SEARCH TRANSACTION

The search pair-list is currently under discussion in [WV-CSP]; a full example cannot be provided.

7.8.1 SearchRequest primitive (1st)

```
WV11SR761 SI=im.user.com#48815@server.com SP=(UL,Smith) SL=5
```

7.8.2 SearchResponse primitive (1st)

```
WV11RS761 SI=im.user.com#48815@server.com SF=7 CF=F SX=5
SD=112233
SR=(wv:john,wv:matthias@salamander.com,wv:francisco@don.com,wv:
mary@site.org,wv:driver@car.org)
```

7.8.3 SearchRequest primitive (continued)

```
WV11SR762 SI=im.user.com#48815@server.com SD=112233 SX=6
```

7.8.4 SearchResponse primitive (continued)

```
WV11RS762 SI=im.user.com#48815@server.com SF=7 CF=T SX=7
SR=(wv:henry@home.se,wv:judas@bigfish.com)
```

7.9 STOP SEARCH TRANSACTIONS

7.9.1 StopSearchRequest primitive

WV11SS763 SI=im.user.com#48815@server.com SD=112233

7.9.2 Status primitive

WV11ST763 SI=im.user.com#48815@server.com ST=200

7.10 INVITATION TRANSACTIONS

7.10.1 InviteRequest primitive

WV11IR761 SI=im.user.com#48815@server.com II=11 IT=PR
PS=(OS,LT,FT) UI=(wv:lara.naval@secret.gov,wv:francisco)
IR="Feel free to use my presence infos!" SN=((matthias,wv:
/chatroup@wv.com)) ON=(john,wv:/chatroup@wv.com)

7.10.2 Status primitive

WV11ST761 ST=200

7.10.3 InviteUserRequest primitive

WV11IU762 SI=54321 II=11 IT=PR SN=((john,wv:/chatroup@wv.com))
PS=(OS,LT,FT) IR="Feel free to use my presence infos!"

7.10.4 Status primitive

WV11ST762 SI=54321 ST=200

7.10.5 InviteUserResponse primitive

WV11UI763 SI=54321 II=11 AC=T IX="Thanks a lot!"

7.10.6 Status primitive

WV11ST763 SI=54321 ST=200

7.10.7 InviteResponse primitive

WV11RI764 SI=im.user.com#48815@server.com II=11 UI=wv:francisco
AC=T IX="Thanks a lot!"

7.10.8 Status primitive

WV11ST764 ST=200

7.11 CANCELING INVITATION TRANSACTIONS

7.11.1 CancelInviteRequest primitive

```
WV11CI761 SI=im.user.com#48815@server.com II=11
  UI=(wv:lara.naval@secret.gov,wv:francisco) RR="I will be on
  vacation for 1 week." ON=(john,wv:/chatroup@wv.com)
```

7.11.2 Status primitive

```
WV11ST761 SI=im.user.com#48815@server.com ST=200
```

7.11.3 CancelInviteUserRequest primitive

```
WV11CU762 SI=54321 II=11 SN=((john,wv:/chatroup@wv.com)) RR="I
  will be on vacation for a week."
```

7.11.4 Status primitive

```
WV11ST762 SI=54321 ST=200
```

7.12 GET LIST OF CONTACT LIST IDS TRANSACTION

7.12.1 GetListRequest primitive

```
WV11GL761 SI=im.user.com#48815@server.com
```

7.12.2 GetListResponse primitive

```
WV11LG761 SI=im.user.com#48815@server.com
  CL=(wv:john/colleagues,wv:john/family, wv:john/friends)
```

7.13 CREATE CONTACT LIST TRANSACTION

7.13.1 CreateListRequest primitive

```
WV11CL761 SI=im.user.com#48815@server.com CL=wv:john/friends
  CP=((DN,"My friends"),(DE,T))
```

7.13.2 Status primitive

```
WV11ST761 SI=im.user.com#48815@server.com ST=(200,"Successfully
  completed.")
```

7.14 DELETE CONTACT LIST TRANSACTION

7.14.1 DeleteListRequest primitive

```
WV11DL761 SI=im.user.com#48815@server.com CL=wv:john/friends
```

7.14.2 Status primitive

```
WV11ST761 SI=im.user.com#48815@server.com ST=(200,"Successfully
completed.")
```

7.15 RETRIEVE A CONTACT LIST TRANSACTION

7.15.1 ListManageRequest primitive

```
WV11LM761 SI=im.user.com#48815@server.com CL=vw:john/friends
```

7.15.2 ListManageResponse primitive

```
WV11ML761 SI=im.user.com#48815@server.com ST=200 CP=((DN,"My
friends"),(DE,T)) UN=((("New
friend",vw:new@friend.org),no.nick@name.com))
```

7.16 ADD USERS TO A CONTACT LIST TRANSACTION

7.16.1 ListManageRequest primitive

```
WV11LM761 SI=im.user.com#48815@server.com CL=vw:john/friends
AN=((("Randall the
Vandal",vw:randall@fairlane.com),no.nick@name.com,(Brainstrom,w
v:bright@dark.com))
```

7.16.2 ListManageResponse primitive

```
WV11ML761 SI=im.user.com#48815@server.com ST=200 UN=((("Randall
the
Vandal",vw:randall@fairlane.com),no.nick@name.com,(Brainstrom,w
v:bright@dark.com),("New friend",vw:new@friend.org))
```

7.17 REMOVE USERS FROM A CONTACT LIST

7.17.1 ListManageRequest primitive

```
WV11LM761 SI=im.user.com#48815@server.com CL=vw:john/friends
RN=((("New friend",vw:new@friend.org))
```

7.17.2 ListManageResponse primitive

```
WV11ML761 SI=im.user.com#48815@server.com ST=200 UN=((("Randall
the
Vandal",vw:randall@fairlane.com),no.nick@name.com,(Brainstrom,w
v:bright@dark.com))
```

7.18 MODIFY PROPERTIES OF CONTACT LIST TRANSACTION

7.18.1 ListManageRequest primitive

```
WV11LM761 SI=im.user.com#48815@server.com CL=vv:john/friends
CP=((DN,"My enemies"),(DE,T))
```

7.18.2 ListManageResponse primitive

```
WV11ML761 SI=im.user.com#48815@server.com ST=200 CP=((DN,"My
enemies"),(DE,T))
```

7.19 CREATE ATTRIBUTE LIST TRANSACTION

7.19.1 CreateAttributeListRequest primitive

```
WV11CA761 SI=im.user.com#48815@server.com PS=(OS,LT,FT)
UI=(vv:matthias@salamander.com,vv:francisco) DL=T
```

7.19.2 Status primitive

```
WV11ST761 SI=im.user.com#48815@server.com ST=(200,"Successfully
completed.")
```

7.20 DELETE ATTRIBUTE LIST TRANSACTION

7.20.1 DeleteAttributeListRequest primitive

```
WV11DA761 SI=im.user.com#48815@server.com CL=vv:john/friends DL=F
```

7.20.2 Status primitive

```
WV11ST761 SI=im.user.com#48815@server.com ST=200
```

7.21 GET ATTRIBUTE LIST(S) TRANSACTION

7.21.1 GetAttributeListRequest primitive

```
WV11GA761 SI=im.user.com#48815@server.com DL=T
```

7.21.2 GetAttributeListResponse primitive

In order to be able to distinguish between User-IDs, Contact-List-IDs and the default attribute list, the parameters are separated into AG, AL, DA.

For clarification the basic syntax of Attribute-Association-List is:

```
(<owner(s)>,<attribute(s)>) ,where
```

<owner(s)>¹ is the contact-list-ID(s) or the user-ID(s) to whom the attributes are assigned.

<attribute(s)> is the name of the presence attribute(s) that are associated.

```
WV11AG761 SI=im.user.com#48815@server.com ST=200
AG=((wv:john/colleagues,OS),(wv:john/family,(OS,FT)))
AL(((wv:matthias@salamander.com,wv:francisco@don.com),(OS,FT))
,(wv:mary@site.com,FT)) DA=OS
```

7.22 SUBSCRIBE/UNSUBSCRIBE PRESENCE TRANSACTION

7.22.1 SubscribePresenceRequest primitive

```
WV11SB761 SI=im.user.com#48815@server.com
UI=(wv:matthias@salamander.com,wv:francisco) CL=wv:john/family
PS=OS
```

7.22.2 Status primitive

```
WV11ST761 SI=im.user.com#48815@server.com ST=(200,"Successfully
completed.")
```

7.22.3 PresenceNotificationRequest primitive

For clarification the basic syntax is:

(<owner(s)>,((<attribute 1>,<value 1>),(<attribute n>,<value n>)) ,where
<owner(s)> is the user-ID(s) to whom the presence attributes and values belong.
<attribute> is the name of the presence attribute to which the value belongs.
<value> is the value of a presence attribute.

```
WV11PN761 SI=im.user.com#48815@server.com
PR=((wv:matthias@salamander.com,((OS,T),(FT,"In the
office"))),(wv:francisco,(OS,T)))
```

7.22.4 Status primitive

```
WV11ST761 SI=im.user.com#48815@server.com ST=200
```

7.22.5 UnsubscribePresenceRequest primitive

```
WV11PS761 SI=im.user.com#48815@server.com
UI=(wv:matthias@salamander.com,wv:francisco)
```

7.22.6 Status primitive

```
WV11ST761 SI=im.user.com#48815@server.com ST=(200,"Successfully
completed.")
```

¹ The default attribute list is presented as "DEFAULT".

7.23 GET WATCHER LIST TRANSACTION

7.23.1 GetWatcherListRequest primitive

```
WV11GW761 SI=im.user.com#48815@server.com
```

7.23.2 GetWatcherListResponse primitive

```
WV11WG761 SI=im.user.com#48815@server.com  
UI=(wv:matthias@salamander.com,wv:francisco)
```

7.24 GET PRESENCE TRANSACTION

7.24.1 GetPresenceRequest primitive

```
WV11GP761 SI=im.user.com#48815@server.com  
UI=(wv:matthias,wv:francisco@don.com) PS=OS
```

7.24.2 GetPresenceResponse primitive

For clarification the basic syntax is:

(<owner(s)>,((<attribute 1>,<value 1>),(<attribute n>,<value n>)) ,where
<owner(s)> is the user-ID(s) to whom the presence attributes and values belong.
<attribute> is the name of the presence attribute to which the value belongs.
<value> is the value of a presence attribute.

```
WV11PG761 SI=im.user.com#48815@server.com ST=200  
PR=((wv:matthias@salamander.com,(OS,T)),(wv:francisco,(OS,T)))
```

7.25 REACTIVE PRESENCE AUTHORIZATION TRANSACTIONS

7.25.1 PresenceAuthRequest primitive

```
WV11PR761 SI=im.user.com#48815@server.com  
UI=wv:matthias@salamander.com PS=(OS,LT,FT)
```

7.25.2 PresenceAuthResponse primitive

```
WV11RP761 SI=im.user.com#48815@server.com  
UI=wv:matthias@salamander.com AC=T
```

7.25.3 CancelAuthRequest primitive

```
WV11CR762 SI=54321 UI=wv:matthias@salamander.com
```

7.25.4 Status primitive

```
WV11ST762 SI=im.user.com#48815@server.com ST=200
```

7.26 UPDATE PRESENCE TRANSACTION

7.26.1 UpdatePresenceRequest primitive

```
WV11UP761 SI=im.user.com#48815@server.com UV=((OS,T),(FT,"In the office"))
```

7.26.2 Status primitive

```
WV11ST761 SI=im.user.com#48815@server.com ST=200
```

7.27 SEND MESSAGE TRANSACTION

Only plain text messages are allowed, thus the MIME-type encoding, size and URI are not carried in the primitive.

7.27.1 SendMessageRequest primitive

```
WV11SM761 SI=im.user.com#48815@server.com DE=T
UI=(wv:matthias@salamander.com,wv:francisco)
GI=wv:/chatroup@wv.com SN=("The boss",wv:/chatroup@wv.com))
MC="Hello everybody! How You guys doing?"
```

7.27.2 SendMessageResponse primitive

```
WV11MS761 SI=im.user.com#48815@server.com ST=(200,"Successfully completed.") MI=11235
```

7.28 PUSHING A MESSAGE FROM THE SERVER TRANSACTION

7.28.1 NewMessage primitive

Only plain text messages are allowed, thus the MIME-type encoding, size and URI are not carried in the primitive.

```
WV11NM761 SI=im.user.com#48815@server.com MI=11235
UI=wv:john@smith.com DT=20011118T1203Z MC="Hello everybody! How
You guys doing?"
```

7.28.2 MessageDelivered primitive

```
WV11MD761 SI=im.user.com#48815@server.com MI=11235
```

7.29 GET MESSAGE LIST TRANSACTION

7.29.1 GetMessageListRequest primitive

```
WV11MR761 SI=im.user.com#48815@server.com GI=wv:/chatroup@wv.com
MN=5
```


7.29.2 GetMessageListResponse primitive

Only message-IDs are carried in the primitive.

```
WV11RM761 SI=im.user.com#48815@server.com
MI=(1212,1123,897,624,372)
```

7.30 RETRIEVING A MESSAGE FROM THE SERVER TRANSACTION

7.30.1 GetMessageRequest primitive

```
WV11GM761 SI=im.user.com#48815@server.com MI=1212
```

7.30.2 GetMessageResponse primitive

Only message-IDs are carried in the primitive.

Only plain text messages are allowed, thus the MIME-type encoding, size and URI are not carried in the primitive.

```
WV11MG761 SI=im.user.com#48815@server.com MI=1212
SN=((Some1,wv:/chatroup@wv.com)) DT=20011118T1203Z MC="Hello
everybody! How You guys doing'?"
```

7.30.3 MessageDelivered primitive

```
WV11NM761 SI=im.user.com#48815@server.com MI=11235
```

7.30.4 Status primitive

```
WV11ST761 SI=im.user.com#48815@server.com ST=200
```

7.31 DELIVERY STATUS REPORT TRANSACTION

7.31.1 DeliveryReportRequest primitive

```
WV11DR761 SI=im.user.com#48815@server.com ST=200 DX=20011118T1204Z
UI=wv:matthias@salamander.com GI=wv:/chatroup@wv.com SN(("The
boss",wv:/othergroup@somewhere.com)) DT=20011118T1203Z MI=11235
```

7.31.2 Status primitive

```
WV11ST761 SI=im.user.com#48815@server.com ST=200
```

7.32 GET BLOCKED USER LIST TRANSACTION

7.32.1 GetBlockedListRequest primitive

```
WV11GB761 SI=im.user.com#48815@server.com
```

7.32.2 GetBlockedListResponse primitive

In order to be able to distinguish between User-IDs, and Group-IDs (and ScreenNames), these are separated into BG, BL and GG, GL and UB, UU, and UG, UL.

```
WV11BG761ab SI=im.user.com#48815@server.com
BL=(wv:he@there.com,wv:she@there.com)
BG=(wv:/chatgroup@nowhere.com,("The
boss",wv:/othergroup@somewhere.com)) BU=T
GL=(wv:matthias@salamander.com,wv:francisco) G
```

```
WV11BG761bb
G=(wv:/rock@roll.com,(Talkative,wv:/nowhere@there.org)) GU=F
```

7.33 BLOCK ENTITY TRANSACTION

7.33.1 BlockEntityRequest primitive

```
WV11BE761 SI=im.user.com#48815@server.com
GL=(wv:mary@site.com,wv:wife@home.org) BU=F GU=T
```

7.33.2 Status primitive

```
WV11ST761 SI=im.user.com#48815@server.com ST=(200,"Successfully
completed.")
```

7.34 CREATE GROUP TRANSACTION

7.34.1 CreateGroupRequest primitive

```
WV11CG761ab SI=im.user.com#48815@server.com
GI=wv:/chatgroup@there.com GP=((NM,"Chit chat
group"),(AT,Restricted),(PM,T),(SE,F),(TO,"Family, relation
ships"),(MU
```

```
WV11CG761bb ,30),(WN,"Welcome to my group. Feel free to discuss
about our current topic.") AJ/T))
```

7.34.2 Status primitive

```
WV11ST761 SI=im.user.com#48815@server.com ST=200
```

7.35 DELETE GROUP TRANSACTION

7.35.1 DeleteGroupRequest primitive

```
WV11DG761 SI=im.user.com#48815@server.com
GI=wv:/chatgroup@there.com
```

7.35.2 Status primitive

WV11ST761 SI=im.user.com#48815@server.com ST=200

7.36 JOIN GROUP TRANSACTION

7.36.1 JoinGroupRequest primitive

WV11JG761 SI=im.user.com#48815@server.com
GI=vv:/chatgroup@there.com SN(("==Bart Simpson==
",vv:/chatgroup@there.com)) JR=T

7.36.2 JoinGroupResponse primitive

WV11GJ761 SI=im.user.com#48815@server.com JU=(Matthias,"Francisco
(of the Dons)","Anonymous12") WT="Welcome to WV!"

7.37 USER INITIATED LEAVE GROUP TRANSACTION

7.37.1 LeaveGroupRequest primitive

WV11LU761 SI=im.user.com#48815@server.com
GI=vv:/chatgroup@there.com

7.37.2 LeaveGroupResponse primitive

WV11UL761 SI=im.user.com#48815@server.com ST=200
GI=vv:/chatgroup@there.com

7.38 SERVER INITIATED LEAVE GROUP TRANSACTION

7.38.1 LeaveGroupResponse primitive

WV11UL761 SI=im.user.com#48815@server.com ST=(809,"You have been
rejected from the group." GI=vv:/chatgroup@there.com

7.38.2 Status primitive

WV11ST761 SI=im.user.com#48815@server.com ST=200

7.39 GET GROUP MEMBERS' LIST TRANSACTION

7.39.1 GetGroupMembersRequest primitive

WV11GM761 SI=im.user.com#48815@server.com
GI=vv:/chatgroup@there.com

7.39.2 GetGroupMembersResponse primitive

```
WV11MG761 SI=im.user.com#48815@server.com AD=vw:john@smith.com
MO=(vw:matthias@salamander.com,vw:francisco)
US=(vw:he@there.com,vw:she@there.com)
```

7.40 ADD GROUP MEMBER(S) TRANSACTION

7.40.1 AddGroupMembersRequest primitive

```
WV11AM761 SI=im.user.com#48815@server.com
GI=vw:/chatgroup@there.com UI=(vw:me@home.com,vw:you@there.com)
```

7.40.2 Status primitive

```
WV11ST761 SI=im.user.com#48815@server.com ST=(200,"Successfully
completed.")
```

7.41 REMOVE GROUP MEMBER(S) TRANSACTION

7.41.1 RemoveGroupMembersRequest primitive

```
WV11RM761 SI=im.user.com#48815@server.com
GI=vw:/chatgroup@there.com UI=(vw:me@home.com,vw:you@there.com)
```

7.41.2 Status primitive

```
WV11ST761 SI=im.user.com#48815@server.com ST=200
```

7.42 MEMBER ACCESS RIGHTS TRANSACTION

7.42.1 MemberAccessRequest primitive

```
WV11ME761 SI=im.user.com#48815@server.com
GI=vw:/chatgroup@there.com
AD=(vw:matthias@salamander.com,vw:francisco)
MO=(vw:he@there.com,vw:she@there.com) US=vw:john@smith.com
```

7.42.2 Status primitive

```
WV11ST761 SI=im.user.com#48815@server.com ST=(200,"Successfully
completed.")
```

7.43 MODIFY GROUP PROPERTIES TRANSACTIONS

7.43.1 GetGroupPropsRequest primitive

```
WV11AM761 SI=im.user.com#48815@server.com
GI=vw:/chatgroup@there.com
```

7.43.2 GetGroupPropsResponse primitive

```
WV11CG761ab SI=im.user.com#48815@server.com GP=((NM,"Chit chat
group"),(AT,Restricted),(PM,T),(SE,F),(TO,"Family,
relationships"),(MU,30),(WN,"Welcome to my group. Feel free to
discuss about our current topi
```

```
WV11CG761bb c.")) OP=((PM,T),(PL,Admin),(IM,T),(AJ/F))
```

7.43.3 SetGroupPropsRequest primitive

```
WV11SP762 SI=im.user.com#48815@server.com
GI=wv:/chatgroup@there.com OP=(PM,T)
```

7.43.4 Status primitive

```
WV11ST762 SI=im.user.com#48815@server.com ST=(200,"Successfully
completed.")
```

7.44 REJECTED LIST TRANSACTIONS

7.44.1 RejectListRequest primitive

```
WV11RE761 SI=im.user.com#48815@server.com
GI=wv:/chatgroup@there.com
AU=(wv:he@there.com,wv:she@there.com)
RU=(wv:matthias@salamander.com,wv:francisco)
```

7.44.2 RejectListResponse primitive

```
WV11ER761ac SI=im.user.com#48815@server.com
UI=(wv:he@there.com,wv:she@there.com)
```

7.45 SUBSCRIBE GROUP CHANGE NOTIFICATION TRANSACTION

7.45.1 SubscribeGroupNoticeRequest primitive (get)

```
WV11SU761 SI=im.user.com#48815@server.com
GI=wv:/chatgroup@there.com SU=G
```

7.45.2 SubscribeGroupNoticeResponse primitive

```
WV11US761 SI=im.user.com#48815@server.com SS=F
```

7.45.3 SubscribeGroupNoticeRequest primitive (set)

```
WV11SG762 SI=im.user.com#48815@server.com
GI=wv:/chatgroup@there.com SU=S
```

7.45.4 Status primitive

```
WV11ST762 SI=im.user.com#48815@server.com ST=(200,"Successfully
completed.")
```

7.45.5 Group change notification primitive

```
WV11GG761 SI=im.user.com#48815@server.com
GI=vw:/chatgroup@there.com JU=(Matthias,Anonymous22)
LU=(Matthias,"Francisco (of the Dons)") GP=((AU,8))
OP=((PL,Mod))
```

7.45.6 Status primitive

```
WV11ST761 SI=im.user.com#48815@server.com ST=(200,"Successfully
completed.")
```

7.46 EXAMPLE FOR MULTIPLE TRANSACTIONS

In the following example the client encapsulates 3 transactions into two SMSes. The client:

Responds to a request (Transaction-ID 700),

Requests to join a group (Transaction-ID 701).

Responds to another request (Transaction-ID 702),

1st SMS:

```
WV11ST700 SI=im.user.com#48815@server.com ST=(200,"Successfully
completed.") & WV11JG701ab SI=im.user.com#48815@server.com
GI=vw:/chatgroup@there.com SN=( ("=-
```

2nd SMS:

```
WV11JG701bb Bart Simpson=-",vw:/chatgroup@there.com)) JR=T &
WV11ST702 SI=im.user.com#48815@server.com ST=200
```

8. STATIC CONFORMANCE REQUIREMENT FOR CSP SMS BINDING:

Req#	Description	C-Req	S-Req	Reference
CSPSMS-1	Support fo SMS encoded with UDH	O	O	
CSPSMS-2	When session is started with UDH and the server supports it (CSPSMS-1), all primitives are encoded with UDH during the session	M	M	
CSPSMS-3	Support for SMS encoded without UDH (textual)	O	O	
CSPSMS-4	When session is started without UDH and the server supports it (CSPSMS-3), all primitives are encoded without UDH during the session	M	M	
CSPSMS-5	Support for one SMS message to contain multiple WV messages	O	O	