Over The Air Settings Specification

Approved version: 7.0 12th September 2001

Doc. Number DSS00234-EN

1. Change History

1.0 09-11-99 NMP Approved
Description: Initial version of the Over The Air Settings Specification document.
Compatibility: -

2.0 09-11-99 NMP Approved
Description: New CSD parameters CSD_CALLTYPE and CSD_CALLSPEED added.
Compatibility: -

3.0 09-11-99 NMP Approved
Description: PORT parameters and encodings for PORT values added
Compatibility: -

4.0 09-11-99 NMP Approved
Description: Support for security on/off included and port number description changed
Compatibility: -

5.0 02-12-99 NMP Approved
Description: Several sections rearranged. New characteristics & parameters added: ISP_NAME, PROXY_AUTHNAME, PROXY_AUTHSECRET, BOOKMARK, NAME, URL.
Compatibility: GSM/SMS token has got new value. Several tokens removed

6.0 30-05-00 NMP Approved
Description: Additions to how a homepage is specified in sections 7.4, and 7.6. Example in chapter 14 corrected to reflect changes regarding how homepage is specified. Error in WSP header in example in chapter 14 corrected.
Compatibility: Error in WSP header corrected. "1F" has been added.

6.5 05-12-00 NMP Approved
Description: Support for GPRS and dynamic username and password added. Added parameters PPP_LOGINTYPE, PROXY_LOGINTYPE and GPRS_ACCESSPOINTNAME. Added values GPRS, MANUAL and DYNAMIC. Renamed CSD_AUTHTYPE to PPP_AUTHTYPE, CSD_AUTHNAME to PPP_AUTHNAME and CSD_AUTHSECRET to PPP_AUTHSECRET. Several sections rearranged.
Compatibility: Some elements have been renamed but their token values have not changed and there should then be no compatibility issues.

6.6 21-03-01 NMP Approved
Description: Support for a CHARACTERISTIC element with TYPE=MMSURL, which can be used to indicate the URL of a MMSC. Furthermore GPRS_ACCESSPOINTNAME is not marked as required in table 1 anymore.
Compatibility: New token for TYPE=MMSURL added. GPRS_ACCESSPOINTNAME is not required for GRPS bearers anymore.

7.0 12-09-01 NMP Approved
Description: Support for a SyncML Settings added.
Compatibility: -

2. Contents

3. Glossary

For the purpose of this specification the following abbreviations apply.

ABNF	Augmented Baskus-Naur Notation
AP	Access Point
HTTP	HyperText Transfer Protocol
IANA	Internet Assigned Numbers Authority
IAP	Internet Access Point
OTA	Over The Air
XML	Extensible Markup Language
WBXML	WAP Binary Extensible Markup Language
DTD	Document Type Definition
WAP	Wireless Application Protocol
WDP	Wireless Datagram Protocol
WSP	Wireless Session Protocol
WTLS	Wireless Transport Layer Security
SMS	Short Message Service
SyncML	Synchronization Markup Language
CSD	Circuit Switched Data
URI	Universal Resource Indicator
URL	Universal Resource Locator
USSD	Unstructured Supplementary Services Data
GPRS	General Packet Radio Service
PPP	Point-to-Point Protocol
URL	Uniform Resource Locater
MIME	Multipurpose Internet Mail Extensions
MMS	MultiMedia Message Service
MMSC	MultiMedia Message Service Center

4. Introduction

This document specifies how to provide mobile phones with browser and SyncML settings over the air. The evolution of the specification is depicted in "Section 1. Change History". This section also indicates to what extent the various versions are backward compatible from a specification viewpoint. Backward compatibility has been considered starting from version 4.3. Regarding compliance statements for specific mobile phone products, please refer to corresponding vendor-specific specifications, if applicable.

The document is structured as follows:

- Glossary
- Introduction
  A brief introduction to the document (this section).
- General description
  Contains a general description of how settings are provided to handsets.
- Description of the XML Document Type Definition for browser settings (DTD).
- Description of the XML document with its elements and attributes for browser settings.
- Description of the WBXML encoding of the XML document for browser settings.
- Description of the XML Document Type Definition for SyncML settings (DTD).
- Description of the elements of the SyncML settings.
- Description of the WBXML encoding of the XML document for SyncML settings.

- References
  List of references related to the specification.
- Appendix – Examples
  Various examples of settings documents with binary encoding examples.

5. General description

In order to make handsets accept browser and SyncML settings sent over the air, the settings must be provided in a binary encoded XML document with a specific MIME-type depending on the setting type. The settings must be pushed over SMS to a predefined WDP port as a WSP connection less unsecure push. This predefined WDP port for browser settings is 49999 and for SyncML settings 49996. Handsets able to receive settings over the air must always listen on this port. The different setting types are described below.

5.1. Setting types

Three (3) setting types are defined each with different purpose, they are:

- Browser settings
- Browser bookmarks
- SyncML settings

5.1.1. Browser settings

The browser settings are used to provide handsets with basic settings needed to establish a connection to be used for browsing. These settings may also include bookmarks. Browser settings are identified by the MIME-type **application/x-wap-prov.browser-settings**.

5.1.2. Browser bookmarks

The browser bookmarks are used to provide handsets with bookmarks of any kind that can be used for browsing. Browser bookmarks are identified by the MIME-type **application/x-wap-prov.browser-bookmarks**.

5.1.3. SyncML settings

The SyncML settings are used to provide handsets with basic settings needed to establish a sync session with a sync server. These settings may also include authentication information. SyncML settings are identified by the MIME-types **application/x-prov.syncset+xml** (clear text XML) or **application/x-prov.syncset+wbxml** (binary XML). The content type code for binary media type is 0x020B.

The support for binary XML is mandatory.

The security model of the SyncML settings format is based on the WAP Provisioning Bootstrap security model [6]. The purpose is to authenticate the sender of the settings. The security information can be specified by including the security information in the MIME header as specified in Chapter 12.

6. Description of the XML DTD for browser settings

```
<!ELEMENT CHARACTERISTIC-LIST (CHARACTERISTIC)+>

<!ELEMENT CHARACTERISTIC (PARM*)>
<!ATTLIST CHARACTERISTIC
  TYPE        CDATA              #REQUIRED
  VALUE       CDATA              #IMPLIED
>

<!ELEMENT PARM EMPTY>
```

```
<!ATTLIST PARM
  NAME      CDATA            #REQUIRED
  VALUE     CDATA            #REQUIRED
>
```

7. Description of the XML document for browser settings

Following is shown a simple example of a XML document:

```
<?xml version="1.0"?>
<!DOCTYPE CHARACTERISTIC-LIST SYSTEM "/DTD/characteristic_list.xml">
<CHARACTERISTIC-LIST>
        <CHARACTERISTIC TYPE="ADDRESS">
                <PARM NAME="BEARER" VALUE="GSM/CSD"/>
                <PARM NAME="PROXY" VALUE="192.122.10.120"/>
                <PARM NAME="CSD_DIALSTRING" VALUE="+358508124002"/>
                <PARM NAME="PPP_AUTHTYPE" VALUE="PAP"/>
                <PARM NAME="PPP_AUTHNAME" VALUE="wapuser"/>
                <PARM NAME="PPP_AUTHSECRET" VALUE="wappassw"/>
        </CHARACTERISTIC>
        <CHARACTERISTIC TYPE="NAME">
                <PARM NAME="NAME" VALUE="Mobilbank Settings"/>
        </CHARACTERISTIC>
        <CHARACTERISTIC TYPE="URL" VALUE="http://wap.dk"/>
        <CHARACTERISTIC TYPE="BOOKMARK">
                <PARM NAME="NAME" VALUE="Mobilbank"/>
                <PARM NAME="URL" VALUE=" http://wap.dk "/>
        </CHARACTERISTIC>
</CHARACTERISTIC-LIST>
```

In the following sections the different elements and attributes of a XML document is described.

7.1. The CHARACTERISTIC element

This element groups the browser settings into logical units of five (5) different types; ADDRESS, URL, NAME, ID, MMSURL and BOOKMARK,

The types ADDRESS, URL, NAME and ID are used when specifying WAP settings and MMSURL are used to specify the URL to the MMSC. The type BOOKMARK are used when specifiying bookmarks

7.1.1. General restrictions or rules concerning CHARACTERISTIC elements

Restrictions or rules related to a specific type is described in the sections related to the type.

7.2. The PARM element

The PARM element is used to provide the actual values for the individual settings parameters within each CHARACTERISTIC element.

7.2.1. General restrictions or rules concerning PARM elements

It is allowed to have the same PARM element within a CHARACTERISTIC type more than once but only the PARM element listed first will be used.

A PARM element can only be coded either as an inline string or as a single byte token.

If a PARM element is marked as required (see table 1) the element is needed in order to make the settings set useful. In general it is recommended to include all the required elements, but if a required element is missing handling of the settings set is manufacturer specific, but is could be one of the solutions below:

1) The entire settings set is discarded
2) Parts of the settings set is discarded
3) The settings set is accepted, the user is then required to add the missing parts manually when taking the set into use.

Please refer to the manufacturer specific support documents for clarification. If an optional PARM element is missing from the list or has a value the handset does not comply with, it is set to a default value if possible.

If a PARM element is coded as a single byte token the default value is shown underlined or is specified in exception cases.

If a PARM element is coded as inline string the default values is empty string.

All parameter lengths specified are recommended maximum sizes. Any manufacturer specific deviations from these values are specified in the manufacturer specific support specifications.

7.3. CHARACTERISTIC elements with TYPE=ADDRESS

Characteristics elements with the TYPE=ADDRESS attribute embrace settings concerning a particular bearer, e.g., GSM SMS or GSM CSD. Several address settings can be provided in one document. However, for each bearer, only the address settings listed first will be used. The type of the bearer is specified by a PARM attribute and depending on the bearer additional PARM elements are required or optional.

7.3.1. BEARER

The PARM element with the NAME=BEARER attribute is used to identify the bearer to be used for a specific setting set. VALUE can be assigned following:

VALUE -> (GSM/CSD | GSM/SMS | GSM/USSD | IS-136/CSD|GPRS)
(See WBXML attribute values defines in section 8.2.)

GSM/CSD is using Circuit Switched Data as bearer, GSM/SMS is using Short Message Service as bearer, GSM/USSD is using Unstructured Supplementary Services Data as bearer and IS-136/CSD is the name for the Circuit Switched Bearer used in the TDMA system. GPRS is the name for the bearer for General Packet Radio Service. The table below shows the association between the different bearers and parameters.

CHARACTERISTIC TYPE	PARM NAME	PARM VALUE	PARM Associated	Required
ADDRESS	BEARER	GSM/CSD	PROXY	√
			PORT	
			CSD_DIALSTRING	√
			CSD_CALLTYPE	
			CSD_CALLSPEED	
			PPP_AUTHTYPE	
			PPP_AUTHNAME	
			PPP_AUTHSECRET	
			PPP_LOGINTYPE	
			PROXY_AUTHNAME	
			PROXY_AUTHSECRET	
			PROXY_LOGINTYPE	
			ISP_NAME	
		GSM/SMS	PROXY	√
			PORT	
			SMS_SMSC_ADDRESS	√

CHARACTERISTIC TYPE	PARM NAME	PARM VALUE	PARM Associated	Required
		GSM/USSD	PROXY_TYPE	√
			PROXY	√
			PORT	
			USSD_SERVICE_CODE	√
		IS-136/CSD	PROXY	√
			PORT	
			CSD_DIALSTRING	√
			PPP_AUTHTYPE	
			PPP_AUTHNAME	
			PPP_AUTHSECRET	
			PPP_LOGINTYPE	
			PROXY_AUTHNAME	
			PROXY_AUTHSECRET	
			PROXY_LOGINTYPE	
			ISP_NAME	
		GPRS	PROXY	√
			PORT	
			GPRS_ACCESSPOINTNAME	
			PPP_AUTHTYPE	
			PPP_AUTHNAME	
			PPP_AUTHSECRET	
			PPP_LOGINTYPE	
			PROXY_AUTHNAME	
			PROXY_AUTHSECRET	
			PROXY_LOGINTYPE	
			ISP_NAME	

Table 1 Association between bearers and parameters

7.3.2. PPP_AUTHTYPE

The PARM element with the NAME=PPP_AUTHTYPE attribute indicates which protocol to use for user authentication. VALUE can be assigned following:

VALUE -> (PAP | CHAP | MS_CHAP)
(See WBXML attribute values defines in section 8.2.)

PAP is short for Password Authentication Protocol, a type of authentication which uses clear-text passwords and is the least sophisticated authentication protocol, and CHAP stands for Challenge Handshake Authentication Protocol, a protocol used to negotiate the most secure form of encrypted authentication supported by both server and client. MS-CHAP (Microsoft(TM)-CHAP) is similar to the CHAP protocol, but is using an encryption scheme that is alternative to the one used for CHAP.

7.3.3. PPP_AUTHNAME

The PARM element with the NAME=PPP_AUTHNAME attribute indicates the login name to be used. VALUE can be assigned following:

VALUE -> login name (using inline string)

Maximum length of login name is 32 bytes.

7.3.4. PPP_AUTHSECRET

The PARM element with the NAME=PPP_AUTHSECRET attribute indicates the password/Secret to be used with the selected authentication protocol. VALUE can be assigned following:

VALUE -> password (using inline string)

Maximum length of password is 20 bytes.

7.3.5. PPP_LOGINTYPE

The PARM element with the NAME=PPP_LOGINTYPE attribute specifies whether a automatic or manual login should be performed in the PPP negotiation at the access point of the service provider.

VALUE -> (AUTOMATIC | MANUAL)
(See WBXML attribute values defines in section 8.2.)

Using the MANUAL logintype the user will be prompted for username and password when a browse session is started.

Using the AUTOMATIC logintype the user will be not be prompted for username and password when a browse session is started, but a static name and password from the WAP settingsset will be used.

7.3.6. PROXY

The PARM element with the NAME=PROXY attribute is used to identify the IP address of the WAP proxy in case of CSD and the service number in case of SMS. In case of USSD the PROXY can be either an IP address or an MSISDN number. This is indicated in the PROXY_TYPE PARM element. VALUE can be assigned following:

VALUE -> proxy (using inline string)

Maximum length of proxy is 21 bytes.

7.3.7. PROXY_TYPE

The PARM element with the NAME=PROXY_TYPE attribute is used to identify the format of the PROXY PARM element. VALUE can be assigned following:

VALUE -> (MSISDN_NO | IPV4)
(See WBXML attribute values defines in section 8.2.)

If PROXY_TYPE=MSISDN_NO the PROXY is a MSISDN number, if PROXY_TYPE=IPV4 the PROXY is an IP address.

PROXY_TYPE is only valid in GSM/USSD settings.

7.3.8. PROXY_AUTHNAME

The PARM element with the NAME=PROXY_AUTHNAME attribute indicates the login name to be used for gateway required authentication. Support of this PARM element is manufacturer specific. VALUE can be assigned following:

VALUE -> login name (using inline string)

Maximum length of login name is 32 bytes.

7.3.9. PROXY_AUTHSECRET

The PARM element with the NAME=PROXY_AUTHSECRET attribute indicates the password/secret to be used for gateway required authentication. Support of this PARM element is manufacturer specific. VALUE can be assigned following:

VALUE -> password (using inline string)

Maximum length of password is 20 bytes.

7.3.10. PROXY_LOGINTYPE

The PARM element with the NAME=PROXY_LOGINTYPE attribute specifies whether a automatic or manual login should be performed at the proxy.

VALUE -> (AUTOMATIC | MANUAL)
(See WBXML attribute values defines in section 8.2.)

Using the MANUAL logintype the user will be prompted for username and password when a browse session is started.

Using the AUTOMATIC logintype the user will be not be prompted for username and password when a browse session is started, but a static name and password from the WAP settingsset will be used.

7.3.11. PORT

The PARM element with the NAME=PORT attribute specifies whether connection less or connection oriented connections should be used VALUE can be assigned following:

VALUE -> (<u>9200</u> | 9201 | 9202 | 9203)
(See WBXML attribute values defines in section 8.2.)

Use 9200 (or 9202) for connection less connections and 9201 (or 9203) for connection oriented connections.

If no PORT is specified, a default value of 9200 is assumed.

Note that port numbers 9202 and 9203 enable secure connections (by means of WTLS), whereas port numbers 9200 and 9201 disable secure connections.

7.3.12. CSD_DIALSTRING

The PARM element with the NAME=CSD_DIALSTRING attribute specifies the MSISDN number of the modem pool. VALUE can be assigned following:

VALUE -> msisdn_number (using inline string)

Maximum length of msisdn_number is 21 bytes.

7.3.13. CSD_CALLTYPE

The PARM element with the NAME=CSD_CALLTYPE attribute indicates the type of circuit switched call to be used for the connection. VALUE can be assigned following:

VALUE -> (<u>ANALOGUE</u> | ISDN)
(See WBXML attribute values defines in section 8.2.)

In general the call type should be set to ANALOGUE since ISDN is not generally available on all networks.

7.3.14. CSD_CALLSPEED

The PARM element with the NAME=CSD_CALLSPEED attribute indicates the desired call speed to be used for the connection. VALUE can be assigned following

VALUE -> (AUTO | 9600 | 14400 | 19200 | 28800 | 38400 | 43200 | 57600)
(See WBXML attribute values defines in section 8.2.)

Default value is AUTO when CSD_CALLTYPE is Analogue and 9600 when CSD_CALLTYPE is ISDN.

7.3.15. ISP_NAME

The PARM element with the NAME=ISP_NAME attribute indicates the name of the Internet Service Provider. Support of this PARM element is manufacturer specific. VALUE can be assigned following:

VALUE -> isp_name (using inline string)

Maximum length of isp_name is 20 bytes.

7.3.16. SMS_SMSC_ADDRESS

The PARM element with the NAME=SMS_SMSC_ADDRESS attribute indicates the MSISDN number of the SMS Service Centre. VALUE can be assigned following:

VALUE -> sms_smsc_address (using inline string)

Maximum length of sms_smsc_address is 21 bytes.

7.3.17. USSD_SERVICE_CODE

The PARM element with the NAME=USSD_SERVICE_CODE attribute indicates the USSD service code. VALUE can be assigned following:

VALUE -> ussd_service_code (using inline string)

Maximum length of ussd_service_code is 10 bytes.

7.3.18. GPRS_ACCESSPOINTNAME

The PARM element with the NAME=GPRS_ACCESSPOINTNAME attribute indicates the Access point name on Gateway GPRS Support Nodes (GGSN). Allowed characters: 'a'-'z', 'A'-'Z', '0'-'9', '.', '-' and '*'

VALUE -> access_point_name (using inline string)

Maximum length of access_point_name is 100 bytes.

7.3.19. Example

```
<CHARACTERISTIC TYPE="ADDRESS">
        <PARM NAME="BEARER" VALUE="GSM/CSD"/>
        <PARM NAME="PROXY" VALUE="192.122.10.120"/>
        <PARM NAME="CSD_DIALSTRING" VALUE="+358508124002"/>
        <PARM NAME="PPP_AUTHTYPE" VALUE="PAP"/>
        <PARM NAME="PPP_AUTHNAME" VALUE="wapuser"/>
        <PARM NAME="PPP_AUTHSECRET" VALUE="wappassw"/>
</CHARACTERISTIC>
```

7.4. CHARACTERISTIC elements with TYPE=URL or TYPE=MMSURL

The CHARACTERISTIC element with the TYPE=URL attribute has only one attribute which indicates the URL of the home page (TYPE=URL) or the URL of the MMSC (TYPE=MMSURL). VALUE can be assigned following:

VALUE -> url (using inline string)

Maximum length of URL is 100 bytes.

7.4.1. Example

```
<CHARACTERISTIC TYPE="URL" VALUE="operator.com"/>
```

See section 7.6 which also describes an element required when specifying a homepage.

7.5. CHARACTERISTIC elements with TYPE=NAME

This element type must contain exactly one PARM element with NAME=NAME, which states the user-recognisable name to apply for the settings. The VALUE of the PARM element can be assigned following:

VALUE -> name (using inline string)

Maximum length of name is 20 bytes.

7.5.1. Example

```
<CHARACTERISTIC TYPE="NAME">
        <PARM NAME="NAME" VALUE="Operator"/>
</CHARACTERISTIC>
```

7.6. CHARACTERISTIC elements with TYPE=BOOKMARK

This element type must contain exactly two PARM elements, which define the name and URL for a homepage or for bookmarks.

When this element is used with the MIME-type **application/x-wap-prov.browser-settings** the first element indicates the homepage to be used together with the corresponding settings. Note that the URL included in this element and the CHARACTERISTIC element TYPE=URL are both required to define a homepage and their content must be equal, see section 7.4. A homepage and several bookmarks can be provided in one document of the MIME-type referred to above. However, the maximum number of bookmarks accepted is manufacturer specific.

When this element is used with the MIME-type **application/x-wap-prov.browser-bookmarks** the element indicates bookmarks only.

7.6.1. NAME

The PARM element with the NAME=NAME attribute indicates the name of the bookmark or homepage. VALUE can be assigned following:

VALUE -> bookmark_name (using inline string)

Maximum length of bookmark_name is 50 bytes.

7.6.2. URL

The PARM element with the NAME=URL attribute indicates the URL of the bookmark or homepage. VALUE can be assigned following:

VALUE -> bookmark_url (using inline string)

Maximum length of bookmark_url is 255 bytes.

7.6.3. Example

```
<CHARACTERISTIC TYPE="BOOKMARK">
        <PARM NAME="NAME" VALUE="Wap page"/>
        <PARM NAME="URL" VALUE="wap.dk"/>
</CHARACTERISTIC>
```

7.7. CHARACTERISTIC elements with TYPE=ID

This element type must contain exactly one PARM element, which defines an ID to be used to provide some security to the provisioning application. The ID should be known by the subscriber through the subscription or through other communication with the operator. When provisioning data containing the ID is received the user is able to verify the received ID with the ID previously received by other means from the operator. Support of this CHARACTERISTIC element is manufacturer specific.

7.7.1. NAME

The PARM element with the NAME=NAME attribute indicates the ID. VALUE can be assigned following:

VALUE -> id (using inline string)

Maximum length of id is 8 bytes.

7.7.2. Example

```
<CHARACTERISTIC TYPE="ID">
        <PARM NAME="NAME" VALUE="12345678"/>
</CHARACTERISTIC>
```

8. Description of the WBXML document for browser settings

The XML document is binary encoded according to WBXML, [1] "WAP Binary XML Content Format", with the following allowed Global Tokens.

Global Token name	Token value
END	0x01
STR_I	0x03

Table 2 Allowed Global Tokens

Numerical values for the individual tokens are given below.

8.1. Tag tokens

All tokens are defined for code page 0.

Token name	Token value
CHARACTERISTIC-LIST	0x05
CHARACTERISTIC	0x06

PARM	0x07

Table 3 Tag Tokens

8.2. Attribute start tokens

Attribute name	Attribute value prefix	Attribute value
TYPE	ADDRESS	0x06
TYPE	URL	0x07
TYPE	NAME	0x08
NAME		0x10
VALUE		0x11
NAME	BEARER	0x12
NAME	PROXY	0x13
NAME	PORT	0x14
NAME	NAME	0x15
NAME	PROXY_TYPE	0x16
NAME	URL	0x17
NAME	PROXY_AUTHNAME	0x18
NAME	PROXY_AUTHSECRET	0x19
NAME	SMS_SMSC_ADDRESS	0x1A
NAME	USSD_SERVICE_CODE	0x1B
NAME	GPRS_ACCESSPOINTNAME	0x1C
NAME	PPP_LOGINTYPE	0x1D
NAME	PROXY_LOGINTYPE	0x1E
NAME	CSD_DIALSTRING	0x21
NAME	CSD_CALLTYPE	0x28
NAME	CSD_CALLSPEED	0x29
NAME	PPP_AUTHTYPE	0x22
NAME	PPP_AUTHNAME	0x23
NAME	PPP_AUTHSECRET	0x24
VALUE	GSM/CSD	0x45
VALUE	GSM/SMS	0x46
VALUE	GSM/USSD	0x47
VALUE	IS-136/CSD	0x48
VALUE	GPRS	0x49
VALUE	9200	0x60
VALUE	9201	0x61

VALUE	9202	0x62
VALUE	9203	0x63
VALUE	AUTOMATIC	0x64
VALUE	MANUAL	0x65
VALUE	AUTO	0x6A
VALUE	9600	0x6B
VALUE	14400	0x6C
VALUE	19200	0x6D
VALUE	28800	0x6E
VALUE	38400	0x6F
VALUE	PAP	0x70
VALUE	CHAP	0x71
VALUE	ANALOGUE	0x72
VALUE	ISDN	0x73
VALUE	43200	0x74
VALUE	57600	0x75
VALUE	MSISDN_NO	0x76
VALUE	IPV4	0x77
VALUE	MS_CHAP	0x78
TYPE	MMSURL	0x7C
TYPE	ID	0x7D
NAME	ISP_NAME	0x7E
TYPE	BOOKMARK	0x7F

Table 4 Attribute Start Tokens

9. Description of the XML DTD for SyncML settings

The elements in the SyncML settings must be sent in a same order as the content models of this DTD elements define them.

```
<!ELEMENT SyncSettings (Version, HostAddr, Port?, RemoteDB+, Name?, Auth*,  ConRef?)>
<!ELEMENT Version (#PCDATA)>
<!ELEMENT HostAddr (#PCDATA)>
<!ELEMENT Port (#PCDATA)>
<!ELEMENT RemoteDB ((CTType, CTVer*)+, URI, Name?, Auth?)>
<!ELEMENT CTType (#PCDATA)>
<!ELEMENT CTVer (#PCDATA)>
<!ELEMENT URI (#PCDATA)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Auth (AuthLevel?, AuthScheme, ((Username | Cred)| (Username, Cred)))>
<!ELEMENT AuthLevel (#PCDATA)>
<!ELEMENT AuthScheme (#PCDATA)>
```

```
<!ELEMENT Username (#PCDATA)>
<!ELEMENT Cred (#PCDATA)>
<!ELEMENT ConRef (ConType, ((Bearer?, AddrType, Addr) | RefID))>
<!ELEMENT ConType (#PCDATA)>
<!ELEMENT Bearer (#PCDATA)>
<!ELEMENT AddrType (#PCDATA)>
<!ELEMENT Addr (#PCDATA)>
<!ELEMENT RefID (#PCDATA)>
```

10. Description of the elements of the SyncML settings

10.1. Addr

Content Model: `(#PCDATA)`

This element specifies the address value of the connection reference. The address can be e.g. phone number or IP address. The type of the address is dependent on the address type defined in the AddrType element.

The support of this element is optional.

10.2. AddrType

Content Model: `(#PCDATA)`

This element specifies the address type value of the connection reference. The value is numeric and encoded as inline string, as specified in the table below.

Address Type	Value	Description
IPV4	'1'	An IPv4 address [10] represented in decimal format with dots as delimiters
IPV6	'2'	An IPv6 address [11] represented as hexadecimal numbers with colons as delimiters or as a combination of hexadecimal and decimal numbers with dots and colons as delimiters
E164	'3'	A phone number according to the E164 scheme
ALPHA	'4'	Generic alphanumeric address
APN	‘5’	Access Point Name

The support of this element is optional.

10.3. Auth

Content Model: `(AuthLevel?, AuthScheme, (Username | Cred | (Username, Cred)))`

This element specifies authentication procedures and credentials used in a SyncML session. This element can be used to authenticate the transport (HTTP, WAP etc.) or the SyncML level. If this element is omitted, authentication is not pre-configured in any level. This element contains information about the authentication level, authentication scheme, username and credentials.

The support of this element is mandatory inside the SyncBody element, but is optional inside the RemoteDB element.

10.4. AuthLevel

Content Model: `(#PCDATA)`

This element specifies the authentication level. The value is numeric and encoded as inline string, as specified in the table below.

Authentication level	Value	Description
SyncML – Server	'1'	Used in the credentials in SyncHdr
HTTP / WSP	'2'	HTTP level authentication
OBEX	'3'	OBEX authentication

This element must be specified if the Auth element is used inside the SyncSettings element, but this element must not be specified if the Auth element is used inside the RemoteDB element.

The support of this element is mandatory.

10.5. AuthScheme

Content Model: `(#PCDATA)`

This element specifies the scheme of the authentication. The value is numeric and encoded as inline string, as specified in the table below.

Authentication scheme	Value
Basic	'1'
Digest (MD5)	'2'

The support of this element is mandatory.

10.6. Bearer

Content Model: `(#PCDATA)`

This element specifies the bearer type of the connection reference. The value is numeric and encoded as inline string, as specified in the table below.

Bearer type	Value
OBEX	‘1’
GSM-USSD	'2'
GSM-SMS	'3'
ANSI-136-GUTS	‘4’
IS-95-CDMA-SMS	‘5’
IS-95-CDMA-CSD	‘6’
IS-95-CDMA-PACKET	‘7’
ANSI-136-CSD	‘8’
ANSI-136-PACKET	‘9’
GSM-CSD	‘10’
GSM-GPRS	‘11’
AMPS-CDPD	‘12’
PDC-CSD	‘13’

PDC-PACKET	‘14’
IDEN-SMS	‘15’
IDEN-CSD	‘16’
IDEN-PACKET	‘17’
FLEX/REFLEX	‘18’
PHS-SMS	‘19’
PHS-CSD	‘20’
TETRA-SDS	‘21’
TETRA-PACKET	‘22’
MOBITEX MPAK	'23’
ANSI-136-GHOST	'24’

The support of this element is optional.

10.7. ConRef

Content Model: `(ConType, ((Bearer?, AddrType, Addr) | RefID))`

This element specifies the reference to a physical or logical access point used by these SyncML settings. This element contains the ConType, Bearer, AddrType, Addr and RefID elements which refer to an access point defined in the device to which this settings document is transferred.

The support of this element is optional.

10.8. ConType

Content Model: `(#PCDATA)`

This element specifies the type of connection. The value is numeric and encoded as inline string, as specified in the table below.

Connection type	Value	Description
WAP 1.x proxy	'1'	-
HTTP / WAP 2.x (WAP NG) proxy	'2'	-
Logical AP	'3'	Name by which access points have been grouped in the configuration
Physical AP	'4'	Refers to a real address: bearer (if needed), address type, address value

If the connection type is WAP 1.x proxy, HTTP / WAP 2.x proxy or Logical AP, the RefID is used to define the connection reference. If the Physical AP is used the Bearer, AddrType, Addr elements define the connection reference.

The support of this element is optional.

10.9. Cred

Content Model: `(#PCDATA)`

This element specifies the credentials of the authentication. The AuthLevel element specifies for which authentication level the given credentials are meant. If the authentication scheme is Basic or Digest the credentials contains the password for the userid specified in the Username element. The credential value must be Base64 character encoded as well encoded as inline string inside the element. The maximum length is 80 bytes after the base64 coding (the cred can be max 60 bytes).

The support of this element is mandatory.

10.10. CTType

Content Model: `(#PCDATA)`

This element specifies the supported media content type of the remote database. The possible values for this element are specified in the IANA web pages [9] and in the SyncML representation protocol [8]. Value for this element can be e.g., text/x-vcard, text/vcard, text/x-vcalendar, or text/calendar. The value is encoded as inline string.

The support of this element is mandatory.

10.11. CTVer

Content Model: `(#PCDATA)`

This element specifies the version of the supported content type. In case the content type i.e. MIME type has several different versions this element needs to be used (e.g. text/html 1.1/1.2/1.3). The formal MIME version content is 1*DIGIT "." 1*DIGIT (e.g. “1.0”), but if MIME versions use different way of indicating the version it can also be used in this element. If this element is not specified, the default value is “1.0” or the lowest version of the MIME. The value is encoded as inline string.

The support of this element is mandatory.

10.12. HostAddr

Content Model: `(#PCDATA)`

This element contains the host address of the SyncML service. This can be either an IP address or a URL. The value is encoded as inline string and the maximum length is 150 bytes. The TCP port number of the origin server (SyncML service) must not be defined in the HostAddr element. The value of the element is specified as LocURI in SyncHdr (see SyncML Representation protocol [8]).

The support of this element is mandatory.

10.13. Name

Content Model: `(#PCDATA)`

This element specifies the displayable name of the SyncML Settings or the remote database. If the element is used in the root element then it specifies the name of the SyncML settings and if it is used inside the RemoteDB element it specifies the displayable name of a remote database. The value is encoded as inline string and the maximum length is 50 bytes. This information can also be shown to a user.

The support of this element is optional.

10.14. Port

Content Model: `(#PCDATA)`

This element specifies the transport port of the origin server (SyncML service). The value is encoded as inline string and the maximum length is 5 bytes.

The support of this element is mandatory. If the element is not specified, the default port value is 80.

10.15. RefID

Content Model: `(#PCDATA)`

This element specifies the logical reference value of the connection reference. The reference value is the logical name or ID of the connection point definition. Characters are case sensitive. The value is encoded as inline string and the maximum length is 20 bytes.

The support of this element is optional.

10.16. RemoteDB

Content Model: `(URI,(CTType, CTVer*)+, Name?, Auth?)`

This element specifies the remote database to be synchronized and the authentication information of the database.

The support of this element is mandatory.

10.17. SyncSettings

Content Model: `(Version, HostAddr, Port?, RemoteDB+, Name?, Auth*, ConRef?)`

This element is the root element.

10.18. URI

Content Model: `(#PCDATA)`

This element specifies the relative or absolute URI of the remote database. This URI is used inside the SyncML Alert and Sync commands. The value is encoded as inline string and the maximum length is 100 bytes.

To specify a certain condition for the database (e.g. the database is classified as a public database) it can be done by using CGI scripting, as specified in SyncML Representation Protocol specification [8]. The possible parameter values can be found from SyncML Representation Protocol specification [8] Section 4.18. In addition to the possible parameter values specified in the SyncML Representation protocol the Calendar database can be specified to support only To-Do's or Events by specified the TODO or EVENT string after the questionmark. See also the CGI scripting example in Section 14.3.

The support of this element is mandatory, but the support of the CGI scripting it is optional.

10.19. Username

Content Model: `(#PCDATA)`

This element specifies the username for the SyncML service or the transport authentication. The AuthLevel element specifies for which authentication level the given username is meant. The value is encoded as inline string and the maximum length is 80 bytes.

The support of this element is mandatory.

10.20. Version

Content Model: `(#PCDATA)`

This element specifies the version of the used SyncML client settings format. Major reversions of the specification create incompatible changes that will generally require a new parser. Minor revisions involve changes that do not impact basic compatibility of the parser. When the XML document conforms to this revision of the SyncML client settings specification the value must be '1.0'. The value is encoded as inline string.

The support of this element is mandatory.

11. Description of the WBXML document for SyncML settings

The following tables define the token assignments for the element types into WBXML as defined by [1].

The WBXML document can only contain inline strings, i.e. usage of opaque data is not allowed.

Element type name	WBXML token (hex value)
Addr	05
AddrType	06
Auth	07
AuthLevel	08
AuthScheme	09
Bearer	0A
ConRef	0B
ConType	0C
Cred	0D
CTType	0E
CTVer	0F
HostAddr	10
Name	11
Port	12
RefID	13
RemoteDB	14
SyncSettings	15
URI	16
Username	17
Version	18

12. Description of the MIME header for SyncML settings

The SyncML Settings MIME header contains the following parameters:

charset Parameter

Purpose: Specifies the character set used to represent the SyncML Settings Format document. The default character set for SyncML Settings Format document is UTF-8, as defined by RFC 2279.

Formal Specification: The following ABNF defines the syntax for the parameter.

chrset-param = ";" "charset" "=" <any IANA registered charset identifier>

version Parameter

Purpose: Specifies the major/minor revision identifiers for the SyncML Settings Format specification that defines the SyncML Settings MIME media type. If present, must be the same value as that specified by the "Version" element type in the SyncML Settings MIME content information. If not present, the default value "1.0" is to be assumed.

Formal Specification: The following ABNF defines the syntax for the parameter.

version-param = ";" "version" "=" 1*numeric "." 1*numeric

text = 1*ALPHA

numeric = "0" / "1" / "2" / "3" / "4" / "5" / "6" / "7" / "8"/ "9"

sec Parmeter

Purpose: Specifies the security mechanism used. If the element is not present, no security is used. If present it must take one of the values USERPIN, USERPINMAC, NETWPIN, USERNETWPIN, as specified in the WAP Provisioning Bootstrap Specification [6]. If the mac parameter is provided, the sec parameter must also be present. There is no default value.

The parameter can have the following values:

Meaning	Value
NETWPIN	0
USERPIN	1
USERNETWPIN	2
USERPINMAC	3

Formal Specification: The following ABNF defines the syntax for the parameter.

sec-param = ";" "sec" "=" 1*numeric

numeric = "0" / "1" / "2" / "3"

mac Parmeter

Purpose: The parameter contains an even number of (upper case) hexadecimal digits used to authenticate the sender of the document and ensure integrity of the document, as defined in the WAP Provisioning Content specification [7]. The calculation of the MAC is defined in the WAP Provisioning Bootstrap Specification [6]. There is no default value.

Formal Specification: The following ABNF defines the syntax for the parameter.

mac-param = ";" "mac" "=" text

13. References

[1] "WAP Binary XML Content Format", WAP Forum, Version 1.3. 16-Jun-1999. URL: http://www.wapforum.org/

[2] "Extensible Markup Language (XML) 1.0", W3C Recommendation 06-October-2000. URL: http://www.w3.org/TR/REC-xml

[3] "Wireless Session Protocol Specification", WAP Forum, Version 28-May-1999. URL: http://www.wapforum.org/

[4] "Wireless Datagram Protocol Specification", WAP Forum, Version 14-May-1999. URL: http://www.wapforum.org/

[5] "IANA MIBEnum Character Set Registry"
URL: ftp://ftp.isi.edu/in-notes/iana/assignments/character-sets

[6] WAP Provisioning Bootstrap Specification ((WAP-184-ProvBoot-20010314-a), WAP Forum

[7] WAP Provisioning Content Specification (WAP-183_001-ProvCont-20010614-p), WAP Forum

[8] SyncML Representation Protocol specification version 1.0, SyncML

[9] IANA Media Types, IANA

[10] Internet Protocol: Darpa internet protocol program specification (RFC791), IETF

[11] Internet Protocol Version 6 addressing architecture (RFC2373), IETF

14. Appendix A – Examples

In the two first examples the reference to the DOCTYPE is shown for illustrative purpose.

14.1. GSM/CSD Settings set

MIME-type used: **application/x-wap-prov.browser-settings**.

14.1.1. XML Document content

```
<?xml version="1.0"?>
<!DOCTYPE CHARACTERISTIC-LIST SYSTEM "file://c:/settingspush/settings.dtd" >
<CHARACTERISTIC-LIST>
        <CHARACTERISTIC TYPE="ADDRESS">
                <PARM NAME="BEARER" VALUE="GSM/CSD"/>
                <PARM NAME="PROXY" VALUE="123.34.6.7"/>
                <PARM NAME="CSD_DIALSTRING" VALUE="+45"/>
                <PARM NAME="PPP_AUTHTYPE" VALUE="PAP"/>
        </CHARACTERISTIC>
        <CHARACTERISTIC TYPE="URL" VALUE=" http://wap.dk "/>
        <CHARACTERISTIC TYPE="NAME">
                <PARM NAME="NAME" VALUE="ABC"/>
        </CHARACTERISTIC>
        <CHARACTERISTIC TYPE="BOOKMARK">
                <PARM NAME="NAME" VALUE="Wap"/>
                <PARM NAME="URL" VALUE=" http://wap.dk "/>
        </CHARACTERISTIC>
</CHARACTERISTIC-LIST>
```

14.1.2. Binary encoding of XML document

SMS CONTENT – SMS 1/2

Binary value	Meaning	Description
0B	User-Data-Header (UDHL) Length = 11 bytes	WDP layer (start WDP headers).
05	UDH IE identifier: Port numbers	
04	UDH port number IE length	
C3	Destination port (high)	Port number 49999
4F	Destinating port (low)	
C0	Originating port (high)	
02	Originating port (low)	
00	UDH IE identifier: SAR	
03	UDH SAR IE length	
04	Datagram ref no.	
02	Total no. of segments in datagram	
01	Segment count	WDP layer (end WDP headers)

01	Transaction ID / Push ID	WSP layer (start WSP headers)
06	PDU type (push)	
2C	Headerslength (content type+headers)	
1F		Length greater than 30
2A	value length (value name not used)	
61,70,70,6C,69,63,61,74,69,6F,6E,2F,78,2D,77,61,70,2D,70,72,6F,76,2E,62,72,6F,77,73,65,72,2D,73,65,74,74,69,6E,67,73	'a','p','p','l','i','c','a','t','i','o','n','/','x','-','w','a','p','-','p','r','o','v','.','b','r','o','w','s','e','r','-','s','e','t','t','i','n','g','s'	MIME-Type = browser settings
00	Null termination of content type string	
81	charset (wellknown PARM.(short integer))	
EA	UTF-8 (using short integer)	WSP layer (end WSP headers)
01	Version	WBXML 1.1
01	Unknown public identifier	
6A	Charset	UTF-8
00	string table length	
45	CHARACTERISTIC_LIST with content	tag
C6	CHARACTERISTIC with content and attributes	tag
06	TYPE=ADDRESS	attribute name with prefix
01	end (attributes)	
87	PARM with attributes	tag
12	NAME=BEARER	attribute name with prefix
45	VALUE=GSM/CSD	attribute name with prefix
01	end (PARM)	
87	PARM with attributes	tag
13	NAME=PROXY	attribute name with prefix
11	VALUE	attribute name
03	Inline string	attribute value
31, 32, 33, 2E, 34, 35, 2E, 36, 2E, 37	'1', '2', '3', '.', '4' ,'5', '.', '6','.', '7'	
00	end inline string	
01	end (PARM)	
87	PARM with attributes	tag
21	NAME=CSD_DIALSTRING	attribute name with prefix
11	VALUE	attribute name
03	inline string	attribute value

2B, 34,35	'+', '4', '5'	
00	end inline string	
01	end (PARM)	
87	PARM with attributes	tag
22	NAME=PPP_AUTHTYPE	attribute name with prefix
70	VALUE=PAP	attribute name with prefix
01	end (PARM)	
01	end (CHARACTERISTIC)	
86	CHARACTERISTIC with attributes	tag
07	TYPE=URL	attribute name with prefix
11	VALUE	attribute name
03	Inline string	attribute value
68, 74, 74, 70, 3A, 2F, 2F, 77, 61, 70, 2E, 64, 6B	'h', 't', 't', 'p', ':', '/', '/', 'w', 'a', 'p', '.', 'd', 'k'	
00	end inline string	
01	end (attributes)	
C6	CHARACTERISTIC with content and attributes	tag
08	TYPE=NAME	attribute name with prefix
01	end attributes	
87	PARM with attributes	tag
15	NAME=NAME	attribute name with prefix
11	VALUE	attribute name
03	inline string	attribute value
41, 42, 43	'A', 'B', 'C'	
00	end inline string	
01	end (PARM)	
01	end (CHARACTERISTIC)	
C6	CHARACTERISTIC with content and attributes	tag
7F	TYPE=BOOKMARK	attribute name with prefix
01	end attributes	
87	PARM with attributes	tag
15	NAME=NAME	attribute name with prefix
11	VALUE	attribute name
03	inline string	attribute value

SMS CONTENT – SMS 2/2

Binary value	Meaning	Description
0B	User-Data-Header (UDHL) Length = 11 bytes	WDP layer (start WDP headers).
05	UDH IE identifier: Port numbers	
04	UDH port number IE length	
C3	Destination port (high)	Port number 49999
4F	Destinating port (low)	
C0	Originating port (high)	
02	Originating port (low)	
00	UDH IE identifier: SAR	
03	UDH SAR IE length	
04	Datagram ref no.	
02	Total no. of segments in datagram	
02	Segment count	WDP layer (end WDP headers)
57, 61, 70	'W', 'a', 'p'	
00	end inline string	
01	end (PARM)	
87	PARM with attributes	tag
17	NAME=URL	attribute name with prefix
11	VALUE	attribute name
03	inline string	attribute value
68, 74, 74, 70, 3A, 2F, 2F, 77, 61, 70, 2E, 64, 6B	'h', 't', 't', 'p', ':', '/', '/', 'w', 'a', 'p', '.', 'd', 'k'	
00	end inline string	
01	end (PARM)	
01	end (CHARACTERISTIC)	
01	end (CHARACTERISTIC_LIST)	

14.2. GSM/GPRS Settings set

MIME-type used: **application/x-wap-prov.browser-settings**.

14.2.1. XML Document content

```
<?xml version="1.0"?>
<!DOCTYPE CHARACTERISTIC-LIST SYSTEM "file://c:/settingspush/settings.dtd" >
<CHARACTERISTIC-LIST>
        <CHARACTERISTIC TYPE="ADDRESS">
                <PARM NAME="BEARER" VALUE="GPRS"/>
                <PARM NAME="PROXY" VALUE="123.45.6.7"/>
                <PARM NAME="GPRS_ACCESSPOINTNAME" VALUE="sonofon.com"/>
                <PARM NAME="PPP_AUTHTYPE" VALUE="PAP"/>
        </CHARACTERISTIC>
        <CHARACTERISTIC TYPE="URL" VALUE=" http://wap.dk "/>
        <CHARACTERISTIC TYPE="NAME">
                <PARM NAME="NAME" VALUE="ABC"/>
        </CHARACTERISTIC>
        <CHARACTERISTIC TYPE="BOOKMARK">
                <PARM NAME="NAME" VALUE="Wap"/>
                <PARM NAME="URL" VALUE=" http://wap.dk "/>
        </CHARACTERISTIC>
</CHARACTERISTIC-LIST>
```

14.2.2. Binary encoding of XML document

SMS CONTENT – SMS 1/2

Binary value	Meaning	Description
0B	User-Data-Header (UDHL) Length = 11 bytes	WDP layer (start WDP headers).
05	UDH IE identifier: Port numbers	
04	UDH port number IE length	
C3	Destination port (high)	Port number 49999
4F	Destinating port (low)	
C0	Originating port (high)	
02	Originating port (low)	
00	UDH IE identifier: SAR	
03	UDH SAR IE length	
04	Datagram ref no.	
02	Total no. of segments in datagram	
01	Segment count	WDP layer (end WDP headers)
01	Transaction ID / Push ID	WSP layer (start WSP headers)
06	PDU type (push)	
2C	Headerslength (content type+headers)	

1F		Length greater than 30
2A	value length (value name not used)	
61,70,70,6C,69,63,61,74,69,6F,6E,2F,78,2D,77,61,70,2D,70,72,6F,76,2E,62,72,6F,77,73,65,72,2D,73,65,74,74,69,6E,67,73	'a','p','p','l','i','c','a','t','i','o','n','/','x','-','w','a','p','-','p','r','o','v','.','b','r','o','w','s','e','r','-','s','e','t','t','i','n','g','s'	MIME-Type = browser settings
00	Null termination of content type string	
81	charset (wellknown PARM.(short integer))	
EA	UTF-8 (using short integer)	WSP layer (end WSP headers)
01	Version	WBXML 1.1
01	Unknown public identifier	
6A	Charset	UTF-8
00	string table length	
45	CHARACTERISTIC_LIST with content	tag
C6	CHARACTERISTIC with content and attributes	tag
06	TYPE=ADDRESS	attribute name with prefix
01	end (attributes)	
87	PARM with attributes	tag
12	NAME=BEARER	attribute name with prefix
49	VALUE=GPRS	attribute name with prefix
01	end (PARM)	
87	PARM with attributes	tag
13	NAME=PROXY	attribute name with prefix
11	VALUE	attribute name
03	Inline string	attribute value
31, 32, 33, 2E, 34, 35, 2E, 36, 2E, 37	'1', '2', '3', '.', '4' ,'5', '.', '6','.', '7'	
00	end inline string	
01	end (PARM)	
87	PARM with attributes	tag
1C	NAME=GPRS_ACCESSPOINTNAME	attribute name with prefix
11	VALUE	attribute name
03	inline string	attribute value
73,6F,6E,6F,66,6F,6E,2E,63,6F,6D	's','o','n','o','f','o','n','.','c','o','m'	
00	end inline string	
01	end (PARM)	
87	PARM with attributes	tag

22	NAME=PPP_AUTHTYPE	attribute name with prefix
70	VALUE=PAP	attribute name with prefix
01	end (PARM)	
01	end (CHARACTERISTIC)	
86	CHARACTERISTIC with attributes	tag
07	TYPE=URL	attribute name with prefix
11	VALUE	attribute name
03	Inline string	attribute value
68, 74, 74, 70, 3A, 2F, 2F, 77, 61, 70, 2E, 64, 6B	'h', 't', 't', 'p', ':', '/', '/', 'w', 'a', 'p', '.', 'd', 'k'	
00	end inline string	
01	end (attributes)	
C6	CHARACTERISTIC with content and attributes	tag
08	TYPE=NAME	attribute name with prefix
01	end attributes	
87	PARM with attributes	tag
15	NAME=NAME	attribute name with prefix
11	VALUE	attribute name
03	inline string	attribute value
41,42,43	'A','B','C'	
00	end inline string	
01	end (PARM)	

SMS CONTENT – SMS 2/2

Binary value	Meaning	Description
0B	User-Data-Header (UDHL) Length = 11 bytes	WDP layer (start WDP headers).
05	UDH IE identifier: Port numbers	
04	UDH port number IE length	
C3	Destination port (high)	Port number 49999
4F	Destinating port (low)	
C0	Originating port (high)	
02	Originating port (low)	
00	UDH IE identifier: SAR	
03	UDH SAR IE length	

04	Datagram ref no.	
02	Total no. of segments in datagram	
02	Segment count	WDP layer (end WDP headers)
01	end (CHARACTERISTIC)	
C6	CHARACTERISTIC with content and attributes	tag
7F	TYPE=BOOKMARK	attribute name with prefix
01	end attributes	
87	PARM with attributes	tag
15	NAME=NAME	attribute name with prefix
11	VALUE	attribute name
03	inline string	attribute value
57, 61, 70	'W', 'a', 'p'	
00	end inline string	
01	end (PARM)	
87	PARM with attributes	tag
17	NAME=URL	attribute name with prefix
11	VALUE	attribute name
03	inline string	attribute value
68, 74, 74, 70, 3A, 2F, 2F, 77, 61, 70, 2E, 64, 6B	'h', 't', 't', 'p', ':', '/', '/', 'w', 'a', 'p', '.', 'd', 'k'	
00	end inline string	
01	end (PARM)	
01	end (CHARACTERISTIC)	
01	end (CHARACTERISTIC_LIST)	

14.3. SyncML Settings set

14.3.1. XML Document content

```
<SyncSettings>
     <Version>1.0</Version>
     <HostAddr>http://www.syncserver.com/sync</HostAddr>
     <Port>8080</Port>
     <RemoteDB>
          <CTType>text/x-vcard</CTType>
          <CTVer>2.1</CTVer>
          <URI>./Contacts?CLASS&EQ;PRIVATE</URI>
          <Name>Private Contact DB</Name>
          <Auth>
               <AuthScheme>1</AuthScheme>
               <Username>james</Username>
               <Cred>cHdk</Cred> <!—Base64 coded ‘pwd’-->
          </Auth>
     </RemoteDB>
     <RemoteDB>
```

```
          <CTType>text/x-vcalendar</CTType>
          <CTVer>1.0</CTVer>
          <URI>./Calendar</URI>
          <Name>Calendar DB</Name>
     </RemoteDB>
     <Name>PIM Service</Name>
     <Auth>
          <AuthLevel>2</AuthLevel>
          <AuthScheme>1</AuthScheme>
          <Username>james</Username>
          <Cred>Ym9uZA==</Cred> <!—Base64 coded ‘bond’-->
     </Auth>
     <Auth>
          <AuthLevel>1</AuthLevel>
          <AuthScheme>1</AuthScheme>
          <Username>bond</Username>
          <Cred>Ym9uZA==</Cred> <!—Base64 coded ‘bond’-->
     </Auth>
     <ConRef>
          <ConType>1</ConType>
          <RefID>My AP</RefID>
     </ConRef>
</SyncSettings>
```

14.3.2. Binary encoding of XML document

The example below shows how the SyncML Settings document is sent over the SMS bearer using the WAP protocols.

SMS CONTENT - SMS 1/3

Binary value	Meaning	Description
0B	User-Data-Header (UDHL) Length = 11 bytes	WDP layer (start WDP headers). See [5]
05	UDH IE identifier: Port numbers	
04	UDH port number IE length	
C3	Destination port (high)	Port number 49996
4C	Destinating port (low)	
C0	Originating port (high)	
02	Originating port (low)	
00	UDH IE identifier: SAR	
03	UDH SAR IE length	
04	Datagram ref no.	
03	Total no. of segments in datagram	
01	Segment count	WDP layer (end WDP headers)
01	Transaction ID / Push ID	WSP layer (start WSP headers) See [6]
06	PDU type (push)	

06	Headerslength (content type+headers)	
05	value length (value name not used)	
02	Short lenght (Well-Known-Media + Long integer used)	
02	Content type code	MIME-Type
0B	Content type code	MIME-Type
81	charset (wellknown PARM.(short integer))	
EA	UTF-8 (using short integer)	WSP layer (end WSP headers)
01	Version	WBXML 1.1
01	Unknown public identifier	
6A	Charset	UTF-8
00	string table length	
55	SyncSettings with content	Tag
58	Version with content	Tag
03	Inline string follows	
31, 2E, 30	String '1.0'	
00	End inline string	
01	End Version element	
50	HostAddr with content	Tag
03	Inline string follows	
68, 74, 74, 70, 3A, 2F, 2F, 77, 77, 77, 2E, 73, 79, 6E, 63, 73, 65, 72, 76, 65, 72, 2E, 63, 6F, 6D, 2F, 73, 79, 6E, 63	String 'http://www.syncserver.com/sync'	
00	End inline string	
01	End HostAddr element	
52	Port with content	Tag
03	Inline string follows	
38, 30, 38, 30	String '8080'	
00	End inline string	
01	End Port element	
54	RemoteDB with content	Tag
4E	CTType with content	Tag
03	Inline string follows	
74, 65, 78, 74, 2F, 78, 2D, 76, 63, 61, 72, 64	String 'text/x-vcard'	
00	End inline string	
01	End CTType element	
4F	CTVer with content	Tag

03	Inline string follows	
32, 2E, 31	String ‘2.1’	
00	End inline string	
01	End CTVer element	
56	URI with content	Tag
03	Inline string follows	
2E, 2F, 43, 6F, 6E, 74, 61, 63, 74, 73, 3F, 43, 4C, 41, 53, 53, 26, 45, 51, 3B, 50, 52, 49, 56, 41, 54, 45	String './Contacts?CLASS&EQ;PRIVATE'	
00	End inline string	
01	End URI element	
51	Name with content	Tag
03	Inline string follows	
50, 72, 69, 76, 61, 74, 65, 20	String ‘Private ‘	

SMS CONTENT - SMS 2/3

Binary value	**Meaning**	**Description**
0B	User-Data-Header (UDHL) Length = 11 bytes	WDP layer (start WDP headers). See [4]
05	UDH IE identifier: Port numbers	
04	UDH port number IE length	
C3	Destination port (high)	Port number 49996
4C	Destinating port (low)	
C0	Originating port (high)	
02	Originating port (low)	
00	UDH IE identifier: SAR	
03	UDH SAR IE length	
04	Datagram ref no.	
03	Total no. of segments in datagram	
02	Segment count	WDP layer (end WDP headers)
43, 6F, 6E, 74, 61, 63, 74, 20, 44, 42	String continues ‘Contact DB’	
00	End inline string	
01	End Name element	

47	Auth with content	Tag
49	AuthScheme with content	Tag
03	Inline string follows	
31	String ‘1’	
00	End inline string	
01	End AuthScheme element	
57	Username with content	Tag
03	Inline string follows	
6A, 61,6D, 65, 73	String ‘james’	
00	End inline string	
01	End Username element	
4D	Cred with content	Tag
03	Inline string follows	
63, 48, 64, 6B	String ‘cHdk’	Base64 coded ‘pwd’
00	End inline string	
01	End Cred element	
01	End Auth element	
01	End RemoteDB element	
54	RemoteDB with content	Tag
4E	CTType with content	Tag
03	Inline string follows	
74, 65, 78, 74, 2F, 78, 2D, 76, 63, 61, 6C, 65, 6E, 64, 61, 72	String 'text/x-vcalendar'	
00	End inline string	
01	End CTType element	
4F	CTVer with content	Tag
03	Inline string follows	
31, 2E, 30	String ‘1.0’	
00	End inline string	
01	End CTVer element	
56	URI with content	Tag
03	Inline string follows	
2E, 2F, 43, 61, 6C, 65, 6E, 64, 61, 72	String './Calendar'	
00	End inline string	
01	End URI element	
51	Name with content	Tag
03	Inline string follows	

43, 61, 6C, 65, 6E, 64, 61, 72, 20, 44, 42	String ‘Calendar DB’	
00	End inline string	
01	End Name element	
01	End RemoteDB element	
51	Name with content	Tag
03	Inline string follows	
50, 49, 4D, 20, 53, 65, 72, 76, 69, 63, 65	String ’PIM Service’	
00	End inline string	
01	End Name element	
47	Auth with content	Tag
48	AuthLevel with content	Tag
03	Inline string follows	
32	String ‘2’	
00	End inline string	
01	End AuthLevel element	
49	AuthScheme with content	Tag
03	Inline string follows	
31	String ‘1’	
00	End inline string	
01	End AuthScheme element	
57	Username with content	Tag
03	Inline string follows	
6A, 61,6D, 65, 73	String ‘james’	

SMS CONTENT - SMS 3/3

Binary value	**Meaning**	**Description**
0B	User-Data-Header (UDHL) Length = 11 bytes	WDP layer (start WDP headers). See [4]
05	UDH IE identifier: Port numbers	
04	UDH port number IE length	
C3	Destination port (high)	Port number 49996
4C	Destinating port (low)	
C0	Originating port (high)	
02	Originating port (low)	
00	UDH IE identifier: SAR	

03	UDH SAR IE length	
04	Datagram ref no.	
03	Total no. of segments in datagram	
03	Segment count	WDP layer (end WDP headers)
00	End inline string	
01	End Username element	
4D	Cred with content	Tag
03	Inline string follows	
59, 6D, 39, 75, 5A, 41, 3D, 3D	String ‘Ym9uZA==’	Base64 coded ‘bond’
00	End inline string	
01	End Cred element	
01	End Auth element	
47	Auth with content	Tag
48	AuthLevel with content	Tag
03	Inline string follows	
31	String ‘1’	
00	End inline string	
01	End AuthLevel element	
49	AuthScheme with content	Tag
03	Inline string follows	
31	String ‘1’	
00	End inline string	
01	End AuthScheme element	
57	Username with content	Tag
03	Inline string follows	
62, 6F, 6E, 64	String ‘bond’	
00	End inline string	
01	End Username element	
4D	Cred with content	Tag
03	Inline string follows	
59, 6D, 39, 75, 5A, 41, 3D, 3D	String ‘Ym9uZA==’	Base64 coded ‘bond’
00	End inline string	
01	End Cred element	
01	End Auth element	
4B	ConRef with content	Tag
4C	ConType with content	Tag
03	Inline string follows	

31	String ‘1’	
00	End inline string	
01	End ConType element	
53	RefID with content	Tag
03	Inline string follows	
4D, 79, 20, 41, 50	String ‘My AP’	
00	End inline string	
01	End RefID element	
01	End ConRef element	
01	End SyncSettings element	