FORUM NOKIA

Series 40 Developer Platform 2.0: OMA Client Provisioning

Version 1.3; December 14, 2004

Series 40

NOKIA

Contents

Change History

March 2004	V1.0	This document replaces *OMA Client Provisioning for Series 40 v1.1*. Appendix B has been modified, application examples have been changed, and FAQ in Appendix C has been added.
July 2004	V1.1	Major restructure; Nokia 3220, future features and Appendix D added
October 14, 2004	V1.2	IM and PoC information in Sections 3.3 and 3.5 updated
December 14, 2004	V1.3	PoC Detail corrections

Disclaimer

The information in this document is provided “as is,” with no warranties whatsoever, including any warranty of merchantability, fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification, or sample. Furthermore, information provided in this document is preliminary, and may be changed substantially prior to final release. This document is provided for informational purposes only.

Nokia Corporation disclaims all liability, including liability for infringement of any proprietary rights, relating to implementation of information presented in this document. Nokia Corporation does not warrant or represent that such use will not infringe such rights.

Nokia Corporation retains the right to make changes to this specification at any time, without notice.

License

A license is hereby granted to download and print a copy of this specification for personal use only. No other license to any other intellectual property rights is granted herein.

1 Introduction

Today's mobile devices contain an increasing number of applications that access resources on the Internet — whether via URLs, gateways, or databases. First application was the browser, which was followed by multimedia messages, where, for example, pictures can be sent to another user, instant messaging allows users to chat, and data synchronization lets owners synchronize their mobile device's contacts and calendar data with an Internet-based database.

The future promises still more options. However, there is already an impressive range of capabilities available to the average user. And in order for users to be able to *use* all of these applications, a large number of *settings* must be in place. This has proven to be a challenge for most users. In response, over-the-air (OTA) provisioning was invented [NOKPROP]. This proprietary solution was able to provision the mobile device with settings, one application at a time.

Open Mobile Alliance (OMA) provisioning [CONTENT, PROVUAB, PROVBOOT] has replaced Nokia's proprietary OTA method [NOKPROP] for the latest products. With OMA, the UI has been generalized to reflect the fact that now, not one, but several applications are being provisioned at a time. This open standard describes how content is formed and sent to the device, and it is extensible parameter-wise, meaning that in the future when new parameters are introduced, present-day devices will continue to work properly. Thus, XML authors don't have to worry about different mobile device versions when composing XML documents.

OMA bootstrap adds security to OMA provisioning in the form of server authentication (see the OMA Bootstrap standard [PROVBOOT]). The major difference between OMA provisioning and the proprietary solution is that it is no longer possible to edit provisioned settings; however, the user can still create and edit his/her own settings (user-created settings). Provisioned settings should be considered the property of the sender (typically a network operator). Also, the device owner can now delete a setting if it has become inoperable or if s/he wants to make room for other settings. Previously it was only possible to replace a setting if the memory was full. In some cases, provisioned sets must first be deleted to make room for new sets.

The following document should be used as a developer's manual for writing XML documents that can be provisioned to Nokia Series 40 Developer Platform 2.0 mobile devices.

2 XML Reference

An OMA provisioning document consists of a number of XML elements defined in [CONTENT]. In general, it consists of a number of *characteristic* elements. Each characteristic element can contain *parm* and nested *characteristic* elements. This chapter offers an overview of the characteristic and parm elements used in different applications running in Nokia Series 40 mobile devices.

2.1 Characteristic: APPLICATION, Proxy, and NAPDef

The use of the elements in the Characteristic: APPLICATION element is strongly dependent on the application type. The application type is specified in the Parm: APPID element and can be, for example, Browser, Multimedia Messaging Service (MMS), Data Synchronization (DS), etc. The different application types and their use of the elements in the Characteristic: APPLICATION element is defined in Client Provisioning Registration documents (PROVREG).

Below is an overview of the characteristic and parm elements used within the Characteristic: APPLICATION element in a Nokia Series 40 device; please refer to PROVREG for further details about Client Provisioning Registration documents.

Characteristic	Type	Nested Characteristic / Parameter		Description	Possible Values	Setting Previously Known As
BOOT-STRAP		NAME		Name of the Configuration Context presented to the user.	E.g., OPERATOR	
		PROVURL		ProvUrl is a key parameter in the OMA Bootstrap security model. This parameter uniquely identifies a Configuration Context. When an OTA-provisioned configuration context is received with the same PROVURL, the user can only replace the existing Configuration Context.		
APPLICA-TION	Browser	APPID		Always “w2” for Browser Application.	• w2	
		TO-NAPID		Link to the network access point: – Defined in same context, must match the NAPID. – If no NAPDEF defined in the same context, value INTERNET shall be used. – Must support GPRS or CSD. Usage of TO-NAPID / TO-PROXY alternatively.	E.g., OPERATOR WAP GPRS or INTERNET	Proxy Disabled
		TO-PROXY		Link to the logical Proxy: – Defined in same context, must match the PROXY-ID. – Must support GPRS or CSD. Usage of TO-NAPID / TO-PROXY alternatively.	E.g., OPERATOR WAP PROXY	Proxy Enabled
		NAME		Name will be visible in UI as account within this CC.	E.g., OPERATOR WAP GPRS	Settings’ name
		RESOURCE	URI	URL of startpage.	E.g., http://www.operator.net	Home-page
			STARTPAGE	Identifies the URL as start page.	No value	

Characteristic	Type	Nested Characteristic / Parameter		Description	Possible Values	Setting Previously Known As
		RESOURCE	URI	URL of an account-specific bookmark. Optional – several bookmarks possible.	E.g., http://www.nokia.com	
			NAME	Name URL of an account-specific bookmark. Optional – several bookmarks possible.	E.g., Nokia.com	
APPLICATION	MMS	APPID		Always "w4" for MMS Application.	• w4	
		TO-NAPID		Link to the Network access point: – Defined in same context, must match the NAPID. – If no NAPDEF defined in the same context, value INTERNET shall be used. – Must support GPRS. Usage of TO-NAPID / TO-PROXY alternatively.	E.g., OPERATOR MMS GPRS or INTERNET	Proxy Disabled
		TO-PROXY		Link to the logical Proxy: – Defined in same context, must match the PROXY-ID. – Must support GPRS. Usage of TO-NAPID / TO-PROXY alternatively.	E.g., OPERATOR MMS PROXY	Proxy Enabled
		NAME		Name will be visible in UI as account within this CC.	E.g., T- D1 MMS	Settings' name
		ADDR		MMS server Address.	E.g., http://mms.operator.net	Home-page
APPLICATION	Wireless Village	APPID		Always "wA" for Wireless Village Application (IM/IMPS).	• wA	
		PROVIDER-ID		Identification of the provider.	E.g., ICQ, AOL, MSN, YHO	
		TO-NAPID		Link to the Network access point: – Defined in same context, must match the NAPID. – If no NAPDEF defined in the same context, value INTERNET shall be used. – Must support GPRS. Usage of TO-NAPID / TO-PROXY alternatively.	E.g., OPERATOR WAP GPRS or INTERNET	Proxy Disabled
		TO-PROXY		Link to the logical Proxy: – Defined in same context, must match the PROXY-ID. – Must support GPRS. Usage of TO-NAPID / TO-PROXY alternatively.	E.g., OPERATOR WAP PROXY	Proxy Enabled
		NAME		Name will be visible in UI as account within the CC.	E.g., WV on MSW	Settings' name
		ADDR		IM and IMPS server address.	E.g., http://www.imps.net:8080/	Home-page
		APPAUTH	AAUTHNAME	User name for authentication on application level.	E.g., wv:first.lastname@nokia.com	UserID
			AAUTHSECRET	User password for authentication on application level.	E.g., 1234	Password

Characteristic	Type	Nested Characteristic / Parameter			Description	Possible Values	Setting Previously Known As
			AAUTHDATA		Private authentication client keys for the server (if needed for this PROVIDER-ID). E.g.: 1. Number of characters in private client key 1 (two digits). 2. Private client key 1. 3. Number of characters in private client key 2 (two digits). 4. Private client key 2.	E.g., "09first_key10second_key"	
APPLICATION	E-Mail SMTP	APPID			Always "25" for e-mail SMTP application.	• 25	
		NAME			Name will be visible in UI as e-mail account within the CC (for all SMTP, POP3, and IMAP with the same PROVIDER-ID).	E.g., EMAILPROV Mailbox	Mailbox name
		PROVIDER-ID			Used to bind SMTP and POP3/IMAP4 settings together. This shall be always used.	E.g., EMAILPROV	
		TO-NAPID			Link to the Network access point: – Defined in same context, must match the NAPID. – If no NAPDEF defined in the same context, value INTERNET shall be used. – Must support GPRS or CSD. Usage of TO-NAPID / TO-PROXY alternatively.	E.g., OPERATOR WAP GPRS or INTERNET	Proxies Disabled
		TO-PROXY			Link to the logical Proxy: – Defined in same context, must match the PROXY-ID. – Must support GPRS or CSD. Usage of TO-NAPID / TO-PROXY alternatively.	E.g., OPERATOR WAP PROXY	Proxies Enabled
		FROM			Specifies own e-mail address. The field name corresponds to FROM field in message. It can also include "My Name" (< and > are guarded < and > around the e-mail address).	E.g., first.last@email.net or myname<first.last@email.net>	E-mail address
		RT-ADDR			Specifies a reply to e-mail address. The field name corresponds to REPLY-TO field in message header.	E.g., first.last@email.net	Reply-to address
		SIGNATURE-INCLUDED			Defines whether SIGNATURE will be included or not.	• OFF • ON	
		SIGNATURE			The actual signature text.		
		APPADDR	ADDR		Specifies the address of sending host.	E.g., mail.email.net	Outgoing (SMTP) server
			PORT	PORTNBR	SMTP port number.	E.g., 25	
		APPAUTH	AAUTHTYPE		Specifies the used authentication mechanism. If AAUTHTYPE is omitted, authentication is disabled.	• CRAM-MD5 • LOGIN • PLAIN	
			AAUTHNAME		SMTP log-in name.	E.g., first.last@email.net	User name
			AAUTHSECRET		SMTP log-in password.	E.g., test1234	Password
APPLICATION	E-Mail POP3	APPID			Always "110" for e-mail SMTP application.	• 110	
		PROVIDER-ID			Used to bind SMTP and POP3 settings together. This shall be always used.	E.g., EMAILPROV	

Characteristic	Type	Nested Characteristic / Parameter	Description	Possible Values	Setting Previously Known As
		MTR	Number of mail items to retrieve (from 1 to 99).	E.g., 30	Retrieve mail
		TO-NAPID	Link to the Network access point: – Defined in same context, must match the NAPID. – If no NAPDEF defined in the same context, value INTERNET shall be used. – Must support GPRS or CSD. Usage of TO-NAPID / TO-PROXY alternatively.	E.g., OPERATOR WAP GPRS	Proxy Disabled
		TO-PROXY	Link to the logical Proxy: – Defined in same context, must match the PROXY-ID. – Must support GPRS or CSD. Usage of TO-NAPID / TO-PROXY alternatively.	E.g., OPERATOR WAP PROXY	Proxy Enabled
		APPADR: ADDR	Specifies the address of receiving POP3 host.	E.g., pop.email.net	Incoming server
		APPADR: PORT: PORTNBR	Port number to connect to receiving POP3 host.	E.g., 110	
		APPAUTH: AAUTHTYPE	Specifies the used authentication mechanism. If AAUTHTYPE is omitted, authentication is disabled.	• CRAM-MD5 • DIGEST-MD5 • LOGIN • PLAIN	
		APPAUTH: AAUTHNAME	Log-in name for POP mailbox.	E.g., first.last@email.net	POP3/ IMAP user name
		APPAUTH: AAUTHSECRET	Log-in password for POP mailbox.	E.g., test1234	POP3/ IMAP password
APPLICATION	E-Mail IMAP4	APPID	Always “143” for e-mail SMTP application.	• 143	
		PROVIDER-ID	Used to bind SMTP and POP3/IMAP4 settings together. This shall be always used.	E.g., EMAILPROV	
		MTR	Number of mail items to retrieve.	E.g., 1 – 99	Retrieve mails
		RM	RETRIEVE-METHOD.	• LATEST • LATEST-UNREAD	Retrieve method
		TO-NAPID	Link to the Network access point: – Defined in same context, must match the NAPID. – If no NAPDEF defined in the same context, value INTERNET shall be used. – Must support GPRS or CSD. Usage of TO-NAPID / TO-PROXY alternatively.	E.g., OPERATOR WAP GPRS or INTERNET	
		TO-PROXY	Link to the logical Proxy: – Defined in same context, must match the PROXY-ID. – Must support GPRS or CSD. Usage of TO-NAPID / TO-PROXY alternatively.	E.g., OPERATOR WAP PROXY	
		APPADR: ADDR	Specifies the address of receiving IMAP4 host.	E.g., imap.email.net	Incoming server
		APPADR: PORT: PORTNBR	Port number to connect to receiving IMAP4 host.	E.g., 143	

Characteristic	Type	Nested Characteristic / Parameter		Description	Possible Values	Setting Previously Known As
		APPAUTH	AAUTHTYPE	Specifies the used authentication mechanism. If AAUTHTYPE is omitted, authentication is disabled.	• LOGIN	
			AAUTHNAME	Log-in name for IMAP mailbox.	E.g., first.last@email.net	POP3/IMAP user name
			AAUTHSECRET	Log-in password for IMAP mailbox.	E.g., test1234	POP3/IMAP password
APPLICATION	SyncML	APPID		Always "w5" for Synchronization Application.	• W5	
		TO-NAPID		Link to the Network access point: – Defined in same context, must match the NAPID. – If no NAPDEF defined in the same context, value INTERNET shall be used. – Must support GPRS. Usage of TO-NAPID / TO-PROXY alternatively.	E.g., OPERATOR MMS GPRS or INTERNET	Proxy Disabled
		TO-PROXY		Link to the logical Proxy: – Defined in same context, must match the PROXY-ID. – Must support GPRS. Usage of TO-NAPID / TO-PROXY alternatively.	E.g., OPERATOR MMS PROXY	Proxy Enabled
		NAME		Name will be visible in UI as account within this CC.	E.g., T- D1 MMS	Settings' name
		ADDR		SyncML Server Address.	E.g., http://www.syn.net/syncml	Home-page
		APPAUT	AAUTHTYPE	Specifies the used authentication mechanism. If omitted, authentication is disabled.	• BASIC • DIGEST	
			AAUTHNAME	Log-in name for Sync Server.	E.g., first.last	User name
			AAUTHSECRET	Log-in password for Sync Server.	E.g., test1234	Password
		RESOURC	URI	Database URI for Contacts.	E.g., CON	
			AACCEPT	Contacts: Specifies the content type the database supports.	• text/x-vcard	
		RESOURCE	URI	Database URI for Calendar (incl. Tasks).	E.g., TCAL	
			AACCEPT	Calendar: Specifies the content type the database supports.	• text/x-vcalendar	
		RESOURC	URI	Database URI for Calendar.	E.g., NOTES	
			AACCEPT	Specifies the content type the database supports.	• text/plain	
PXLOGICAL	Proxy	PROXY-ID		Reference of the Proxy – used within APPLICATION or ACCESS characteristics.	E.g., OPERATOR MMS PROXY	
		NAME		Name of the Proxy.	E.g., OPERATOR MMS PROXY	
		PXPHYSICAL	PHYSICAL-PROXY-ID	ID within this context.	E.g., OPERATOR MMS PX	
			PXADDR	Proxy address.	E.g., 172.28.23.131	Primary proxy
			PXADDRTYPE	Proxy address Type.	• IPV4 • ALPHA	

Characteristic	Type	Nested Characteristic / Parameter	Description	Possible Values	Setting Previously Known As
		PUSHENABLED	Service Loading Push messages allowed (disabled/enabled).	• 0 • 1	
		TO-NAPID	Link to the Network access point – defined in same context: – Defined in same context, must match the NAPID. – If no NAPDEF defined in the same context, value INTERNET shall be used. – Must support GPRS or CSD.	E.g., OPERATOR MMS GPRS	
		PORT: PORTNPR	Port number to connect to PROXY.	E.g., 8008	Primary proxy port
NAPDEF	NAP for GPRS usage	NAPID	Reference ID of this GPRS Network access point: – Used within APPLICATION or PROXY or ACCESS characteristics.	E.g., OPERATOR WAP GPRS	
		NAME	Name of the Network access point – visible in UI.	E.g., OPERATOR WAP GPRS	
		BEARER	The bearer is GSM-GPRS.	• GSM-GPRS	Data bearer GPRS
		NAP-ADDRESS	GPRS access point.	E.g., internet.Operator.net	GPRS access point
		NAP-ADDRTYPE	Address Type.	• APN	
		INTERNET	When this parameter is given, the NAPID can be selected by the user for preferred Internet access. Always define this when applicable.	No value	
		NAPAUTHINFO: AUTHTYPE	Type of authentication. Usage of authentication is optional.	• PAP • CHAP	Authentication type: normal / secure
		NAPAUTHINFO: AUTHNAME	User name for authentication. The usage of user name and password is optional.	E.g., Internet	User name
		NAPAUTHINFO: AUTSECRET	Password for authentication. The usage of user name and password is optional.	E.g., Operator	Password
NAPDEF	NAP for CSD usage	NAPID	Reference ID of this CSD Network access point: – Used within APPLICATION or PROXY or ACCESS characteristics.	E.g., OPERATOR WAP CSD	
		NAME	Name of the Network access point. Visible in UI.	E.g., OPERATOR WAP CSD	
		BEARER	The bearer is GSM-CSD.	• GSM-CSD	Data bearer GSM-Data
		NAP-ADDRESS	CSD Dial-up number.	E.g., 22243	Dial-up number
		NAP-ADDRTYPE	Address Type.	• E164	

Characteristic	Type	Nested Characteristic / Parameter	Description	Possible Values	Setting Previously Known As
		CALLTYPE	Data call type.	• ANALOG-MODEM • V.110	Data call type: analog / ISDN
		LINKSPEED	Data call speed.	• Autobaud-ing • 9,600 • 14,400 • 19,200 • 28,800	Data call speed: Autobaud-ing / 9,600 baud / 14,400 baud
		INTERNET	When this parameter is given, the NAPID can be selected by the user for preferred Internet access. **Always define this when applicable.**	• No value	
		NAPAUTHINFO / AUTHTYPE	Type of authentication. Usage of authentication is optional.	• PAP • CHAP	Authenti-cation type: normal / secure
		NAPAUTHINFO / AUTHNAME	User name for authentication. The usage of user name and password is optional.	E.g., Internet	User name
		NAPAUTHINFO / AUTSECRET	Password for authentication. The usage of user name and password is optional.	E.g., Operator	Password
ACCESS		RULE	Name of access rule.	E.g., General Internet access	
		TO-NAPID	Link to the Network access point: – Defined in same context, must match the NAPID. – Must support GPRS or CSD. Usage of TO-NAPID / TO-PROXY alternatively.	E.g., OPERATOR WAP GPRS or INTERNET	
		TO-PROXY	Link to the logical Proxy: – Defined in same context, must match the PROXY-ID. – Must support GPRS or CSD. Usage of TO-NAPID / TO-PROXY alternatively.	E.g., OPERATOR WAP PROXY	

Characteristic	Type	Nested Characteristic / Parameter	Description	Possible Values	Setting Previously Known As
APPLICA-TION	Push-to-talk Over Cellular	APPID	Always “w9002” for Push-to-talk Over Cellular service.	w9002	
		PROVIDER-ID	Identification of the provider.	E.g., ICQ, AOL, MSN, YHO	
		TO-NAPID	Link to the Network access point: – Defined in same context, must match the NAPID. – If no NAPDEF defined in the same context, value INTERNET shall be used. – Must support GPRS. Usage of TO-NAPID / TO-PROXY alternatively.	E.g., OPERATOR POC GPRS or INTERNET	Proxies Disabled

Characteristic	Type	Nested Characteristic / Parameter		Description	Possible Values	Setting Previously Known As
		TO-PROXY		Link to the Network access point: – Defined in same context, must match the PROXY-ID. – Must support GPRS. Usage of TO-NAPID / TO-PROXY alternatively.	E.g., OPERATOR POC PROXY	Proxies Enabled
		NAME		Name will be visible in UI as account within the CC.	E.g., POC on Operator	Settings' name
		ADDR		Push-to-talk server address URI.	E.g., 121.121.121.121	
		ADDRTYPE		Type of the PoC server address.	IPV4 or IPV6	
		APPAUTH	AAUTHNAME	User name for authentication on application level.	E.g., pttusername@poc.operator.com	UserID
			AAUTHSECRET	User password for authentication on application level.	E.g., pttpassword	Password
			AAUTHDATA	Private authentication client keys for the server (if needed).	E.g., "binary data"	REALM
		RESOURCE	URI	Specifies the URL for the SIP Portal.	E.g., http://sip.operator.com	

Characteristic	Type	Nested Characteristic / Parameter	Description	Possible Values	Setting Previously Known As
APPLICATION	Streaming	APPID	Always “554” 3GPP PSS Streaming using Real-Time Streaming Protocol (RTSP).	554	
		TO-NAPID	Link to the Network access point: – Defined in same context, must match the NAPID. – If no NAPDEF defined in the same context, value INTERNET shall be used. – Must support GPRS. Usage of TO-NAPID / TO-PROXY alternatively.	E.g., OPERATOR STREAMIN GPRS or INTERNET	Proxies Disabled
		TO-PROXY	Link to the Network access point – defined in same context, must match the PROXY-ID – must support GPRS Usage of TO-NAPID / TO-PROXY alternatively.	E.g., OPERATOR STREAMING PROXY	Proxies Enabled
		NAME	Name will be visible in UI as account within the CC.	E.g., STREAMING on Operator	Settings' name
		MIN-UDP-PORT	Minimum UDP port number used for media data traffic (RTP). The default value is product-specific. Value has to be even.	E.g., 6970	Homepage

Characteristic	Type	Nested Characteristic / Parameter	Description	Possible Values	Setting Previously Known As
		MAX-UDP-PORT	Maximum UDP port number used for media data traffic (RTP). The default value is product-specific. Value must be at least MIN-UDP-PORT + 5 to have enough ports for three media streams (audio, video, timed text), preferably much higher.	E.g., 7176	

3 Examples

In this chapter, Sections 3.1 through 3.6 show examples of the XML elements needed to provision the different applications. The first example in Section 3.1 also shows the XML elements in an OMA provisioning document needed for a client to connect to a network, for example the Characteristic: PXLOGICAL and Characteristics: NAPDEF elements. Sections 3.2 through 3.6 only show the Characteristic: APPLICATION element for the relevant application type.

Section 3.8 describes the mobile device's UI when deleting a settings set.

3.1 Browser and Multimedia Messaging Service (MMS) XML Example

```
<?xml version="1.0"?>
<!DOCTYPE wap-provisioningdoc PUBLIC "-//WAPFORUM//DTD PROV 1.0//EN"
"http://www.wapforum.org/DTD/prov.dtd">
<wap-provisioningdoc version="1.0">

      <characteristic type="BOOTSTRAP">
            <parm name="NAME" value="Gnu"/>
      </characteristic>

      <characteristic type="PXLOGICAL">
            <parm name="PROXY-ID" value="timon.dk"/>
            <parm name="NAME" value="Timon Proxy"/>
            <characteristic type="PORT">
                  <parm name="PORTNBR" value="8080"/>
            </characteristic>
            <characteristic type="PXPHYSICAL">
                  <parm name="PHYSICAL-PROXY-ID" value="Timon_Proxy"/>
                  <parm name="PXADDR" value="timon.proxy.dk"/>
                  <parm name="PXADDRTYPE" value="IPV4"/>
                  <parm name="TO-NAPID" value="timon_GPRS"/>
            </characteristic>
      </characteristic>

      <characteristic type="PXLOGICAL">
            <parm name="PROXY-ID" value="gnu.dk"/>
            <parm name="NAME" value="gnu Proxy"/>
            <characteristic type="PORT">
                  <parm name="PORTNBR" value="8080"/>
            </characteristic>
            <characteristic type="PXPHYSICAL">
                  <parm name="PHYSICAL-PROXY-ID" value="gnu Proxy"/>
                  <parm name="PXADDR" value="gnu.proxy.dk"/>
                  <parm name="PXADDRTYPE" value="IPV4"/>
                  <parm name="TO-NAPID" value="gnu_GPRS"/>
                  <parm name="TO-NAPID" value="gnu_CSD"/>
            </characteristic>
      </characteristic>

      <characteristic type="NAPDEF">
            <parm name="NAPID" value="gnu_GPRS"/>
            <parm name="BEARER" value="GSM-GPRS"/>
            <parm name="NAME" value="gnu GPRS"/>
            <parm name="NAP-ADDRESS" value="internet"/>
            <parm name="NAP-ADDRTYPE" value="APN"/>
            <characteristic type="NAPAUTHINFO">
                  <parm name="AUTHTYPE" value="PAP"/>
                  <parm name="AUTHNAME" value="Zimba"/>
                  <parm name="AUTHSECRET" value="Scar"/>
            </characteristic>
      </characteristic>

      <characteristic type="NAPDEF">
            <parm name="NAPID" value="gnu_CSD"/>
            <parm name="BEARER" value="GSM-CSD"/>
```

```
            <parm name="NAME" value="gnu CSD"/>
            <parm name="NAP-ADDRESS" value="+5555555555"/>
            <parm name="NAP-ADDRTYPE" value="E164"/>
            <parm name="CALLTYPE" value="ANALOG-MODEM"/>
            <parm name="LINKSPEED" value="9600"/>
            <characteristic type="NAPAUTHINFO">
                  <parm name="AUTHTYPE" value="PAP"/>
                  <parm name="AUTHNAME" value="Pumba"/>
                  <parm name="AUTHSECRET" value="Sazu"/>
            </characteristic>
      </characteristic>

      <characteristic type="NAPDEF">
            <parm name="NAPID" value="timon_GPRS"/>
            <parm name="BEARER" value="GSM-GPRS"/>
            <parm name="NAME" value="timon GPRS"/>
            <parm name="NAP-ADDRESS" value="internet"/>
            <parm name="NAP-ADDRTYPE" value="APN"/>
            <parm name="INTERNET"/>
            <characteristic type="NAPAUTHINFO">
                  <parm name="AUTHTYPE" value="PAP"/>
                  <parm name="AUTHNAME" value="Rafiki"/>
                  <parm name="AUTHSECRET" value="Kiara"/>
            </characteristic>
      </characteristic>

      <characteristic type="APPLICATION">
            <parm name="APPID" value="w2"/>
            <parm name="TO-PROXY" value="gnu.dk" />
            <parm name="NAME" value="Browser"/>
            <characteristic type="RESOURCE">
                  <parm name="URI" value="http://wap.gnu.dk"/>
                  <parm name="STARTPAGE"/>
            </characteristic>
      </characteristic>

      <characteristic type="APPLICATION">
            <parm name="APPID" value="w4"/>
            <parm name="TO-PROXY" value="timon.dk" />
            <parm name="ADDR" value="http://wap.timon.dk" />
      </characteristic>
</wap-provisioningdoc>
```

3.2 Data Synchronization (DS) XML Example

In this example, only the Characteristics: APPLICATION element is presented. For a working OMA provisioning document, this element must be added to the example in Section 3.1.

```
      <characteristic type="APPLICATION">
            <parm name="APPID" value="w5"/>
            <parm name="TO-PROXY" value="gnu.dk" />
            <parm name="NAME" value="Superman SyncML"/>
            <parm name="ADDR" value="http://metropolis.com/service/sync"/>
            <characteristic type="RESOURCE">
                  <parm name="URI" value="./contacts"/>
                  <parm name="NAME" value="Contacts DB"/>
                  <parm name="AACCEPT" value="text/x-vcard"/>
            </characteristic>
            <characteristic type="RESOURCE">
                  <parm name="URI" value="./calendar"/>
                  <parm name="NAME" value="Calendar DB"/>
                  <parm name="AACCEPT" value="text/x-vcalendar"/>
            </characteristic>
            <characteristic type="RESOURCE">
                  <parm name="URI" value="./notes"/>
```

```
            <parm name="NAME" value="Notes DB"/>
            <parm name="AACCEPT" value="text/plain"/>
      </characteristic>
      <characteristic type="APPAUTH">
            <parm name="AAUTHNAME" value="Clark"/>
            <parm name="AAUTHSECRET" value="Kent"/>
      </characteristic>
</characteristic>
```

3.3 Instant Messaging and Presence Services (IMPS) XML Example

In this example, only the Characteristics: APPLICATION element is presented. For a working OMA provisioning document, this element must be added to the example in Section 3.1.

3.3.1 Wireless Village

Wireless Village is special in two ways. Several accounts are an obvious use case, and then the provisioned username and password need to be changed as well. This enables the use case where several persons can share the same device to access several accounts. This functionality requires that the same server is used for both accounts. The Nokia 3320 device is the first device that supports this.

When Wireless Village settings are created locally at the device, usernames need to be handled in a special way. The device will add the prefix “wv:” automatically to the username. This prefix is not shown in the UI when editing the same parameters, and it’s only relevant for the user-created settings.

The “wv:” prefix is not added to the provisioned username.

If a provisioned username contains the prefix “wv:” and the username is then edited, the “wv:” prefix will not be shown to the end user.

The other special case is multiple accounts. One user can have several accounts to different services. This means, for example, that different chat rooms can be accessed through the same service provider (for example, Sonofon). In addition, the user can select a different account for presence and instant messaging (for example, MSN for presence and AOL for instant messaging).

```
<characteristic type="APPLICATION">
      <parm name="APPID" value="wA"/>
      <parm name="TO-PROXY" value="gnu.dk" />
      <parm name="NAME" value="Cat woman"/>
      <parm name="ADDR" value="http://metropolis.com/service/imps"/>
      <characteristic type="APPAUTH">
            <parm name="AAUTHNAME" value="Hulk"/>
            <parm name="AAUTHSECRET" value="Hogan"/>
      </characteristic>
</characteristic>
```

3.4 3GPP PSS Streaming Using Real-Time Streaming Protocol (RTSP) XML Example

In this example, only the Characteristics: APPLICATION element is presented. For a working OMA provisioning document, this element must be added to the example in Section 3.1.

```
<characteristic type="APPLICATION">
      <parm name="APPID" value="554"/>
      <parm name="NAME" value="Streaming"/>
```

```
      <parm name="TO-NAPID" value="timon_GPRS"/>
      <parm name="MIN-UDP-PORT" value="6970"/>
      <parm name="MAX-UDP-PORT" value="7170"/>
</characteristic>
```

3.5 Push-to-Talk Service (PoC) XML Example

In this example, only the Characteristics: APPLICATION element is presented. For a working OMA provisioning document, this element must be added to the example in Section 3.1.

PoC provisioning is based on PROVREG document. This states that according to the standard the application address can be found in two different places:

- APPLICATION/ADDR
- APPLICATION/APPLADDR/ADDR

APPLICATION/APPLADDR/ADDR is only supported in the following devices:

- The Nokia 5140 mobile device
- The Nokia 7270 imaging device
- The Nokia 6170 imaging device

Furthermore, the PoC clients have three application-specific parameters:

- Nickname
- Domain
- Username

The Nickname, Domain, and Username values aren't provisioned, but they are supposed to be set by the user. In order to maximize usability, the values are set to default by using the APPLICATION/AAUTHNAME value (APPLICATION/AAUTHNAME = xxx@yyy). This is done in the following way:

- Nickname = xxx
- Username = xxx
- Domain = yyy

The value of APPLICATION/AAUTHNAME is not changed during this process. The user can change the default value, but it has no effect on the PoC client's ability to work.

This is done for all Series 40 products except for the Nokia 5140 imaging device.

```
<characteristic type="APPLICATION">
      <parm name="APPID" value="w9002"/>
      <parm name="NAME" value="Push to Talk"/>
      <parm name="TO-NAPID" value="timon_GPRS"/>
      <characteristic type="APPADDR">
            <parm name="ADDR" value="3ffa:1213:3112:c611::74"/>
            <parm name="ADDRTYPE" value="IPv6"/>
            <characteristic type="PORT">
                  <parm name="PORTNBR" value="5060"/>
            </characteristic>
      </characteristic>
```

```
            <characteristic type="APPAUTH">
                  <parm name="AAUTHNAME" value="pttusername@domain"/>
                  <parm name="AAUTHSECRET" value="pttpassword"/>
                  <parm name="AAUTHDATA" value="binary data"/>
            </characteristic>
            <characteristic type="RESOURCE">
                  <parm name="URI" value="http://www.someoperator.com"/>
            </characteristic>
      </characteristic>
```

3.6 E-Mail SMTP, IMAP, and POP3 XML Example

In this example, only the Characteristics: APPLICATION element is presented. For a working OMA provisioning document, this element must be added to the example in Section 3.1.

```
      <characteristic type="APPLICATION">
            <parm name="APPID" value="110"/>
            <parm name="TO-NAPID" value="timon_GPRS"/>
            <parm name="PROVIDER-ID" value=”Mail_Provider”/>
            <characteristic type="APPADDR">
                  <parm name="ADDR" value="pop3.mail.com"/>
                  <characteristic type="PORT">
                        <parm name="PORTNBR" value="110"/>
                        <parm name="SERVICE" value="110"/>
                  </characteristic>
            </characteristic>
            <characteristic type="APPAUTH">
                  <parm name="AAUTHTYPE" value=" CRAM-MD5"/>
                  <parm name="AAUTHNAME" value="pop3username"/>
                  <parm name="AAUTHSECRET" value="pop3password"/>
            </characteristic>
      </characteristic>

      <characteristic type="APPLICATION">
            <parm name="APPID" value="143"/>
            <parm name="TO-NAPID" value="timon_GPRS"/>
            <parm name="PROVIDER-ID" value=”Mail_Provider”/>
            <parm name="MTR" value="10"/>
            <parm name="RM" value="LATEST"/>
            <characteristic type="APPADDR">
                  <parm name="ADDR" value="imap4.mail.com"/>
                  <characteristic type="PORT">
                        <parm name="PORTNBR" value="143"/>
                  </characteristic>
            </characteristic>
            <characteristic type="APPAUTH">
                  <parm name="AAUTHTYPE" value="LOGIN"/>
                  <parm name="AAUTHNAME" value="imap4username"/>
                  <parm name="AAUTHSECRET" value="imap4password"/>
            </characteristic>
</characteristic>
      <characteristic type="APPLICATION">
            <parm name="APPID" value="25"/>
            <parm name="TO-NAPID" value="timon_GPRS"/>
            <parm name="FROM" value="gnu@hotmail.com"/>
            <parm name="NAME" value="Mail Service”/>
            <parm name="PROVIDER-ID" value=”Mail_Provider”/>
            <characteristic type="APPADDR">
                  <parm name="ADDR" value="smtp.mail.com"/>
                  <characteristic type="PORT">
                        <parm name="PORTNBR" value="25"/>
                        <parm name="SERVICE" value="25"/>
                  </characteristic>
            </characteristic>
            <characteristic type="APPAUTH">
```

```
            <parm name="AAUTHTYPE" value="LOGIN"/>
            <parm name="AAUTHNAME" value="smtpusername"/>
            <parm name="AAUTHSECRET" value="smtppassword"/>
        </characteristic>
</characteristic>
```

3.7 Receiving a WBXML Document

As mentioned earlier, only bootstrapped documents can be received over the air. Essentially, OMA Bootstrap is OMA Provisioning plus server authentication. The server authentication is brought about using a *shared secret method*. Two such methods are supported in Series 40 Developer Platform 2.0: the *user pin* method or the *network pin* method.

The shared secret used in the network pin method is the IMSI from the SIM card. In the user pin method, the user manually enters a secret code known only to the user and the provisioning server.

Settings that have been provisioned to the mobile device cannot be edited or viewed, except for some individual user data (user names and passwords).

3.7.1 Save bootstrap set (user pin)

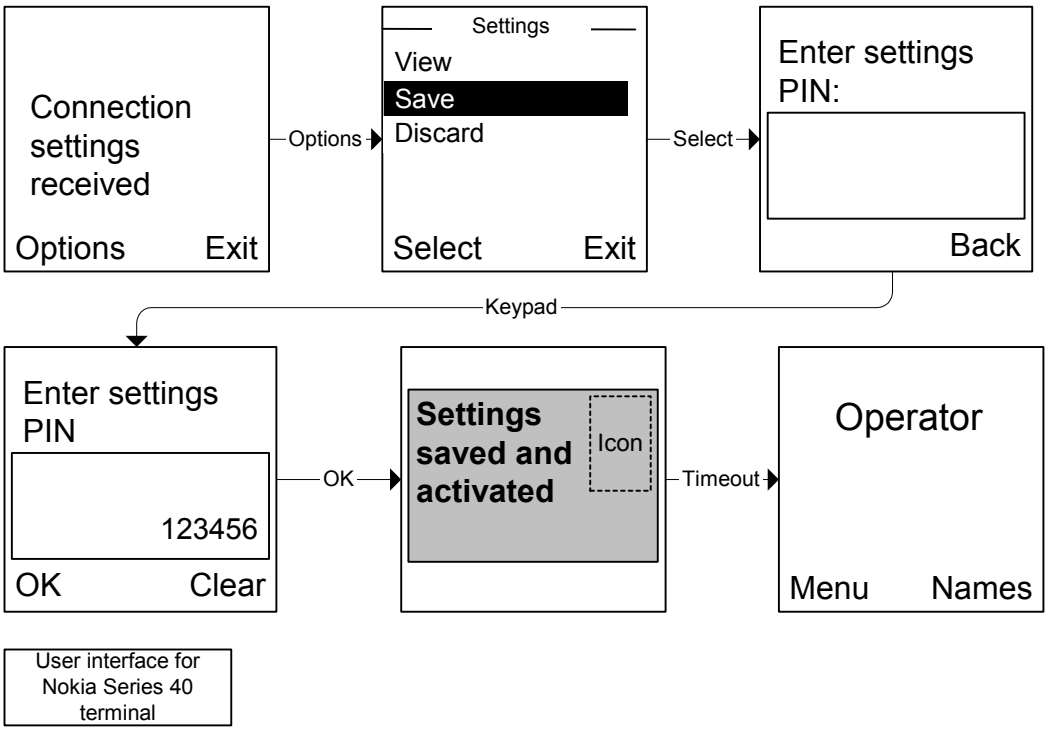

Figure 1: Saving the settings

The pin can be from one to ten digits in length.

3.8 Deleting a Settings Set

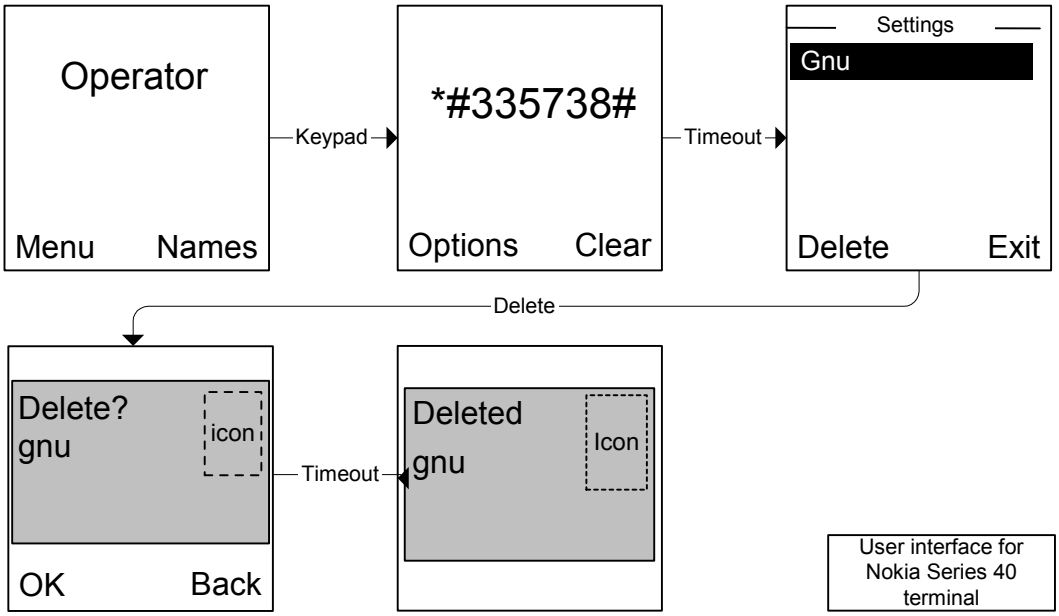

Figure 2: Deleting a setting set

Nokia 3220 device: The user can delete settings using the mobile device's UI (Menu -> Settings -> Configuration Settings -> Default Configuration settings -> Options -> Delete).

4 Client Provisioning Characteristics of the Nokia 3220 Device

By using OMA Client Provisioning, all Series 40 Developer Platform 2.0 applications can be provisioned with connection and application settings. OMA Client Provisioning is standardized by the OMA. It includes two provisioning mechanisms:

- OTA (over the air) – or OMA Bootstrap
- Pre-production

OMA standards specify the priority of those media. The last option has the highest priority and the first has the lowest priority.

4.1 Default Access Points

OMA provisioning includes the concept of a default access point that offers access to the Internet. Thus, it is possible to specify an access point that can be used by any application without prior knowledge to a specific service provider.

This functionality is ensured in the OMA provisioning document, where the settings for the access point are defined. Along with those settings, one INTERNET setting can be set. If this is set, it indicates that the access point supports access to the Internet and may be used accordingly.

The mobile device keeps a list of all provisioned access points, from among which the user can freely choose. If the user doesn't make a selection, then the first provisioned access point is used as the default access point. This will typically be the one from the home operator with the highest priority according to the OMA provisioning document.

In a typical scenario, the operator provides access to the network and the service provider provides access to the mobile services (for example, e-mail or data synchronization services). The operator and the service provider do provisioning individually, which means that they get one Configuration Context apiece. The service provider specifies that an access point will offer access to the Internet and can be used for such access. The mobile device then automatically uses the default access point. Another typical use case is a Java™ MIDlets that accesses the Internet. The last option is about access Internet over HTTP or TCP/IP.

Internet-enabled access points can be used across Configuration Contexts.

4.2 Selection

The OMA provisioning document can, in many cases, hold any number of certain settings (for example, service provider, access points, bearer, and so on). When there are several instances of a certain setting, the first one that appears gets the lowest rank, the next gets a higher rank, and so on.

When a mobile device has no specific rules for making a selection, then the parameter with the lowest rank is always selected. (This is the parameter with the highest priority according to the OMA standard.) A quick look at the bearer illustrates this point. One OMA provisioning document can support many different bearers. The mobile device typically only supports a subset. The device selects the bearer with the lowest rank (highest priority) that is supported.

The concept is that first, the user makes as selection from among the available configuration contexts, and then selects from among the service providers. Furthermore, the user can select the default access point from the access points available for Internet access. There are rules for default selection so that the user doesn't need to make any selections in order to get the services working.

4.3 Well-Formed and Valid Documents

The mobile device performs a check on the received OMA Client Provisioning document according to the specification in the standard. This means it is checked according to the Data Type Definition (DTD). This means that documents can be discarded or parts can be ignored.

It is important to note that the content of the OMA Client Provisioning document can't be checked. For example, it is not possible to check if certain combinations of settings will work in a given environment.

4.4 Create, View, and Edit

The OMA Client Provisioning document holds more than 100 different settings that can be organized in many different combinations.

Ultimately, usability means that the average user will never realize that provisioning had been done. The user will remove the Nokia mobile device from its box and begin using all of the new mobile services that the device supports. The average user does not need to view or edit the provisioned settings.

4.5 Access Rights to Provisioned Settings

By default, the Configuration Context is owned by the initiator of the Configuration Context, which means the operator, the service provider, the local administrator who is using local bootstrap, or the user who is entering settings manually. The default access rights follow the owner. Therefore, the provisioned settings are stored as read-only settings.

4.6 User Names and Passwords

There are many user names and passwords in the provisioning document. The user names and passwords belonging to the Application Characteristic are of special interest because they identify a certain account created by a service provider (for example, they may identify a data synchronization or a Wireless Village user).

User names and passwords are special because they are private by nature. The mobile device encrypts all passwords before saving them. Passwords are decrypted before they are used.

The provisioned settings are:

- Authentication Characteristic
- Authentication Username/Password

If the Authentication Characteristic is included, Username and Password are used, otherwise no authentication is done (for example, typical browser case).

The next rules compile only if the Authentication Characteristic is included:

If Authentication Username/Password is included, then the value is used as Username/Password; otherwise, the user is prompted for Username.

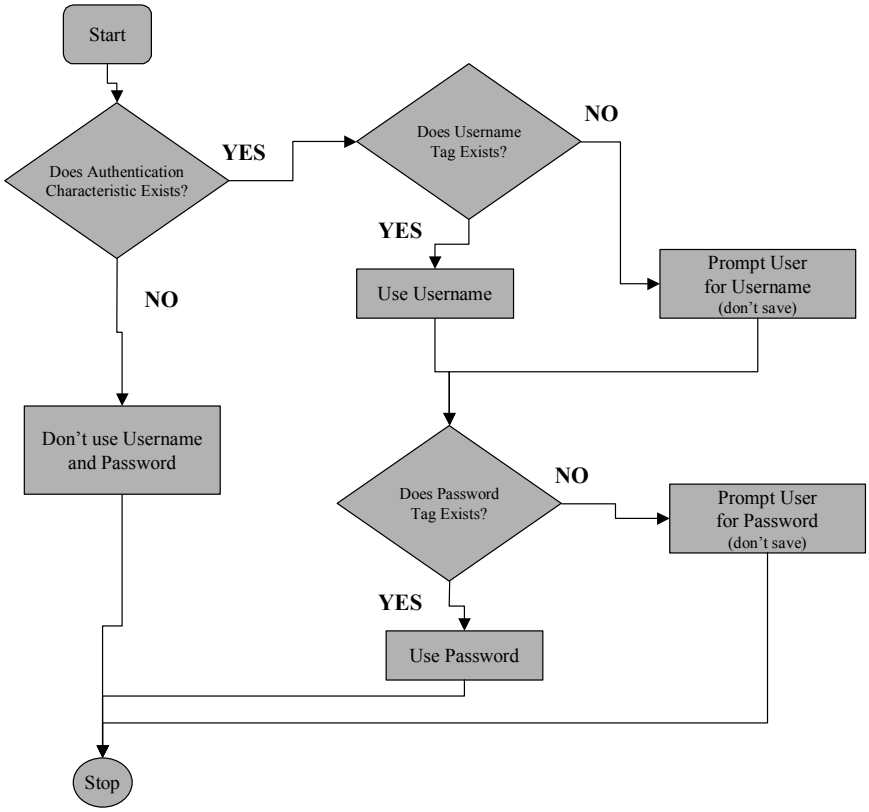

Figure 3: Handling of user names and passwords

Please note that the values can be empty strings, entered values are not stored in the device, and implementation in the Nokia 3220 device only supports two options:

The Authentication Characteristic isn't included and Username and Password are not used.

The Authentication Characteristic is included and Username and Password are expected to be included. This means no prompting for Username and Password

When the user name and password are provisioned (could be empty strings) and when they are viewed for the user, then the user name and password also become editable. When the user name and password are edited, then the edited values are saved as well. These options are offered in addition to those illustrated in Figure 3.

4.7 Configuration Context and Service Accounts

Each service provider owns one Configuration Context. A service provider may offer multiple accounts within one configuration context for one service. This means, for example, that one service provider may offer multiple messaging accounts.

Different accounts are identified by a parameter in the Application Characteristic called the Provider ID. This is a unique URL. Along with the Provider ID, there is also a readable name, which is used to offer the user the option to select from among the accounts.

In general, this ability to select accounts is offered to each application. Selection takes place in two steps:

· 1. The Configuration Context is selected (for example, Sonera).

· 2. The account is selected (for example, Sonera Hot Mail).

If the user doesn't make a selection, the application will use the default Configuration Context and the account with the lowest rank (the first one that appears in the OMA Client Provisioning document).

4.8 Push Proxy Gateway Validation

According to OMA Push Architecture, model push messages are sent from a Push Proxy Gateway (PPG). Certain pushed messages are critical to be ignored unless it has been verified that they have arrived from a trusted PPG. This process is called Push Proxy Gateway Validation.

PPG is provisioned like any other setting. It is a PROXY with a special flag set that indicates that this is a PPG. All incoming push messages with certain MIME types are checked to make sure that they come from trusted source before they are executed. The only MIME that is checked by default is the Service Inbox Service Load (SL).

4.9 Wireless Village

The Wireless Village is special in two ways. First, user can share the same device with other users to access several accounts. This means, of course, that the same server is accessed. The Nokia 3320 device supports this capability.

Second, it allows for multiple accounts. One user can have several accounts to different services. This means, for example, that different chat rooms can be accessed through the same service provider (for example, Sonofon).

4.10 Java(TM) Technology

In most cases, Java(TM) MIDlets only need access to the Internet; this allows them to make HTTP and TCP requests. The default access point is the preferred connection. This means that all MIDlets will be using the same Internet access point; the user is offered a simple UI for changing the default Internet access point.

The implementation supports specific provisioning of a certain MIDlet. This means that a specific MIDlet can be provisioned by specific settings. However, the application ID must be unique. With the Nokia 3320 device, the application ID is created by combining the MIDlet name and vendor:

·"x-midlet-" + "first 20 characters of MIDlet Vendor" + "first 20 characters of MIDlet Name"

Provisioning is done using this application ID. Before creating any connection to the network (using HTTP and/or TCP) the MIDlet checks if any specific settings have been provisioned. If this is the case, those settings are used; if not, the default Internet access point is used.

This rule applies to connection settings as well as application protocol settings. This means that application settings can be provisioned and used by a certain MIDlet.

Internet access is not supported for Proxies. If an operator would like to grant access anyway, it can be achieved in the following way:

- Use the Access characteristic:
  `<characteristic type="ACCESS">`
- Define a rule that all unknown applications should use a specific proxy:
  `<parm name="RULE" value="Default Rule"/>`
- Define a rule that all applications not pointing to a specific access point or proxy should use a "default proxy":
  `<parm name="TO-PROXY" value="10.0.0.172 GPRS"/>`

This approach has a major drawback — it does not work across configuration contexts, and that effectively restricts the number of configuration contexts to one. Note: This option is not recommended at all.

4.11 E-Mail

Several protocols must be available in order for e-mail to work. First of all, different protocols are used for sending and receiving e-mails. Next, there are two different protocols for receiving. The protocol options are:

- SMTP (sending)
- IMAP4 (receiving)
- POP3 (receiving)

All three protocols are used when accessing the e-mail server operated by a service provider. The sending and receiving mail server must be operated by the same provider. The device must pair a certain receiving and sending protocol together. The provider ID is one of the settings available in all of the protocols. This setting is used to pair the sending and receiving protocol.

The user selects the service provider (and thereby also the e-mail server) to use by selecting the SMTP protocol. The mobile device presents a list of providers offering SMTP. Once this selection has been made, the matching receiving e-mail protocols are identified.

Some e-mail service providers offer users a choice between POP3 and IMAP4, and some only offer one. If only one of the receiving protocols is provisioned, then this is automatically selected. If both are provisioned, the user can select one. If no selection is made, IMAP4 is the default.

Service provider identification may be missing in the provisioned documents. The mobile device can handle one set of e-mail protocols without service provider identification. If there is more than one set provisioned, the rest are ignored. The protocols with the lowest rank are used. E-mail provisioning was introduced in Nokia Series 40 devices before the OMA had accepted the following registration documents: 25.txt (SMTP), 110.txt (POP3), and 143.txt (IMAP4). The products that follow Nokia implementation are Nokia 6820, Nokia 6810, Nokia 7200, Nokia 5140, Nokia 6230, and Nokia 7600 devices.

The Nokia 3220 device also supports e-mail, but the registration document has been accepted. Figure 4 lists the names that have been changed.

Registration Documents	Nokia Proprietary Name	OMA Name
143.txt (IMAP4)	APPLICATION/MAIL-TO-RETRIEVE	APPLICATION/MTR
	APPLICATION/RETRIEVE-METHOD	APPLICATION/RM
110.txt (POP3)	APPLICATION/MAIL-TO-RETRIEVE	APPLICATION/MTR
25.txt (SMTP)	APPLICATION/REPLY-TO-ADDRESS	APPLICATION/RT-ADDR

Figure 4: Changed names from Nokia Proprietary to OMA e-mail registration documents

Note: The Nokia 3220 device supports both formats in order to be backward compatible.

5 Client Provisioning Characteristics of Future Nokia Products

By using OMA Client Provisioning, all applications can be provisioned with connection and application settings in Series 40 Developer Platform 2.0. OMA Client Provisioning is standardized by the OMA. It will include four provisioning mechanisms:

- Local (infrared or Bluetooth)
- OTA (over the air) – or OMA Bootstrap
- SC (Smartcard), effectively SIM
- Preproduction

OMA standards specify the priority of those media. The last option has the highest priority and the first has the lowest priority.

Furthermore, the device can be locally bootstrapped over local connectivity (OBEX over Bluetooth or infrared). This allows the user to create his/her own settings and provision them to the device. This feature is not standardized and doesn't contradict existing open standards.

5.1 Smartcard Provisioning

With SC provisioning, OMA Client Provisioning document is stored at the SIM card. Stored document is read during the power-up; it is handled as a normal bootstrap situation and installed in a Configuration Context. The main difference with this Configuration Context is that it takes the default role. Thus, all applications in which the user hasn't selected a specific Configuration Context will automatically start using the new configuration context.

If a preconfigured Configuration Context exists, SC provisioning still takes the default role because it has higher priority. The OMA Client Provisioning document is stored in the WIM/PKCS#15 directory, which requires that the SIM card support storing the configuration document.

The provisioning content can be saved in different directories:

- Bootstrap
- Config1
- Config2

The device checks the directories in the order listed above. When the first file is found, this is used and the rest are not checked. The different directories imply different modes about the conditions for writing files back to the smartcard. The Nokia 3220 device does not support writing to a smartcard.

If a configuration context in the device has the same ProvUrl as in the provisioning document on the smartcard, then the smartcard takes precedence.

The existing configuration context is not deleted and will reappear for the user again if the smartcard is changed.

5.2 Local Bootstrap Using Infrared or Bluetooth

If for some reason the user would like to enter more complex structures, this can be done by the “local bootstrap.” This means creating a provisioning document on a PC and then sending the document to the device over Bluetooth or infrared. This must be done in WBXML. The file must have the extension “prov” (for example, provider-setings.prov), or the OMA Client Provisioning document MIME type must be included.

6 References

CONTENT	*Provisioning Content 1.1*, Open Mobile Alliance, OMA-WAP-ProvCont-v1_1, http://www.openmobilealliance.org/
IANA	*Internet Assigned Numbers Authority*, http://www.iana.org/
PROVBOOT	*Provisioning Bootstrap 1.1*, Open Mobile Alliance, OMA-WAP-PROVBOOT-V1_1, http://www.openmobilealliance.org/
PROVUAB	*Provisioning User Agent Behavior 1.1*, Open Mobile Alliance, OMA-WAP-PROVUAB-V1_1, http://www.openmobilealliance.org/
NOKPROP	*Over The Air Settings Specification*, Version 7.0, http://www.forum.nokia.com/documents
WBXML	*Binary XML Content Format*, WAP Forum, WAP-192-WBXML, http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html
WINA	*WAP Interim Naming Authority*, Open Mobile Alliance, http://www.wapforum.org/wina/
PROVREG	Client Provisioning Registration, http://www.forum.nokia.com/technologies | Device Management & Data Synchronization | OMA Client Provisioning | Documents
WDP	*Wireless Datagram Protocol*, WAP Forum, WAP-259-WDP, http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html
WSP	*Wireless Session Protocol*, WAP Forum, WAP-230-WSP, http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html
GSM 11.11	Digital cellular telecommunications system (Phase 2+) (GSM); Specification of the Subscriber Identity Module - Mobile Equipment (SIM - ME) interface (GSM 11.11 version 7.3.0 Release 1998), http://pda.etsi.org/pda/home.asp?wki_id=8YZC1KFScf5D66E8GvtO
GSM 03.38	Digital cellular telecommunications system (Phase 2+) (GSM); Alphabets and language-specific information (GSM 03.38 version 7.2.0 Release 1998), http://pda.etsi.org/pda/home.asp?wki_id=Hh2gGAJlm@GMOIHmYKNB
GSM 03.40	Digital cellular telecommunications system (Phase 2+) (GSM); Technical Realization of Short Message Service (SMS); Point-to-Point (PP) (GSM 03.40 version 7.2.0 Release 1998), http://pda.etsi.org/pda/home.asp?wki_id=' x nRP'KzAGICDUGG5G

Appendix A. WBXML Binary Example

The following is the WAP Binary XML (WBXML)-encoded version of the XML example presented in Section 3.1. For further details about encoding WBXML documents, see [WBXML] and [CONTENT].

```
030B6A00 C54601C6 56018707 0603476E  ..j..F..V.....Gn
75000101 C6510187 15060374 696D6F6E  u....Q.....timon
2E646B00 01870706 0354696D 6F6E2050  .dk.....Timon P
726F7879 0001C653 01872306 03383038  roxy...S..#..808
30000101 C6520187 2F060354 696D6F6E  0....R../..Timon
5F50726F 78790001 87200603 74696D6F  _Proxy... ..timo
6E2E7072 6F78792E 646B0001 87210685  n.proxy.dk...!..
03000187 22060374 696D6F6E 5F475052  ......timon_GPR
53000101 01C65101 87150603 676E752E  S.....Q.....gnu.
646B0001 87070603 676E7520 50726F78  dk......gnu Prox
790001C6 53018723 06033830 38300001  y...S..#..8080..
01C65201 872F0603 676E7520 50726F78  ..R../..gnu Prox
79000187 20060367 6E752E70 726F7879  y... ..gnu.proxy
2E646B00 01872106 85030001 87220603  .dk...!.........
676E755F 47505253 00018722 0603676E  gnu_GPRS.....gn
755F4353 44000101 C6520187 2F060342  u_CSD....R../..B
69736F6E 2050726F 78790001 87200603  ison Proxy... ..
6269736F 6E2E7072 6F78792E 646B0001  bison.proxy.dk..
87210685 03000187 22060367 6E755F47  .!.........gnu_G
50525300 01872206 03676E75 5F435344  PRS.....gnu_CSD
00010101 01C65501 87110603 676E755F  ......U.....gnu_
47505253 00018710 06AB0300 01870706  GPRS............
03676E75 20475052 53000187 08060369  .gnu GPRS......i
6E746572 6E657400 01870906 89030001  nternet.........
C65A0187 0C069A03 0001870D 06035A69  .Z.............Zi
6D626100 01870E06 03536361 72000101  mba......Scar...
01C65501 87110603 676E755F 43534400  ..U.....gnu_CSD.
01871006 AA030001 87070603 676E7520  .............gnu
43534400 01870806 032B3535 35353535  CSD......+555555
35353535 00018709 06870300 01870D0A  5555............
06900300 01872506 03393630 300001C6  ......%..9600...
5A01870C 069A0300 01870D06 0350756D  Z.............Pum
62610001 870E0603 53617A75 00010101  ba......Sazu....
C6550187 11060374 696D6F6E 5F475052  .U.....timon_GPR
53000187 1006AB03 00018707 06037469  S.............ti
6D6F6E20 47505253 00018708 0603696E  mon GPRS......in
7465726E 65740001 87090689 030001C6  ternet...........
5A01870C 069A0300 01870D06 03526166   Z.............Raf
696B6900 01870E06 034B6961 72610001  iki......Kiara..
0101C600 01550187 36000006 03773200  .....U..6....w2.
01870001 39000006 03676E75 2E646B00  ....9....gnu.dk.
01870706 0342726F 77736572 0001C600  .....Browser....
01590187 3A000006 03687474 703A2F2F  .Y..:....http://
7761702E 676E752E 646B0001 871C0101  wap.gnu.dk......
01C60001 55018736 00000603 77340001  ....U..6....w4..
87000139 00000603 74696D6F 6E2E646B  ...9....timon.dk
00018700 01340000 06036874 74703A2F  .....4....http:/
2F776170 2E74696D 6F6E2E64 6B000101  /wap.timon.dk...
01                                   .
```

WBXML explained:

03	WBXML version (1.3)
0B	Public ID: “-//WAPFORUM//DTD PROV1.0//EN”
6A	Encoding: UTF8
00	String Table - Length
C54601	wap-provisioningdoc version=1.0
C65601	characteristic - BOOTSTRAP
87070603476E750001	parm - NAME, value="Gnu"
01	END (characteristic BOOTSTRAP)
C65101	characteristic - PXLOGICAL
8715060374696D6F6E2E646B0001	parm - PROXY-ID, value="timon.dk"
8707060354696D6F6E2050726F78790001	parm - NAME, value="Timon Proxy"
C65301	characteristic - PORT
87230603383038300001	parm - PORTNBR, value="8080"
01	END (characteristic PORT)
C65201	characteristic - PXPHYSICAL
872F060354696D6F6E5F50726F78790001	parm - PHYSICAL-PROXY-ID, value="Timon_Proxy"
8720060374696D6F6E2E70726F78792E646B0001	parm - PXADDR, value="timon.proxy.dk"
87210685030001	parm - PXADDRTYPE, value=IPV4
8722060374696D6F6E5F475052530001	parm - TO-NAPID, value="timon_GPRS"
01	END (characteristic PXPHYSICAL)
01	END (characteristic PXLOGICAL)
C65101	characteristic - PXLOGICAL
87150603676E752E646B0001	parm - PROXY-ID, value="gnu.dk"
87070603676E752050726F78790001	parm - NAME, value="gnu Proxy"
C65301	characteristic - PORT
87230603383038300001	parm - PORTNBR, value="8080"
01	END (characteristic PORT)
C65201	characteristic - PXPHYSICAL
872F0603676E752050726F78790001	parm - PHYSICAL-PROXY-ID, value="gnu Proxy"
87200603676E752E70726F78792E646B0001	parm - PXADDR, value="gnu.proxy.dk"
87210685030001	parm - PXADDRTYPE, value=IPV4
87220603676E755F475052530001	parm - TO-NAPID, value="gnu_GPRS"
87220603676E755F4353440001	parm - TO-NAPID, value="gnu_CSD"
01	END (characteristic PXPHYSICAL)
C65201	characteristic - PXPHYSICAL
872F06034269736F6E2050726F78790001	parm - PHYSICAL-PROXY-ID, value="Bison Proxy"
872006036269736F6E2E70726F78792E646B0001	parm - PXADDR, value="bison.proxy.dk"
87210685030001	parm - PXADDRTYPE, value=IPV4
87220603676E755F475052530001	parm - TO-NAPID, value="gnu_GPRS"
87220603676E755F435344000101	parm - TO-NAPID, value="gnu_CSD"
01	END (characteristic PXPHYSICAL)
01	END (characteristic PXLOGICAL)
C65501	characteristic - NAPDEF
87110603676E755F475052530001	parm - NAPID, value="gnu_GPRS"
871006AB030001	parm - BEARER, value=GSM-GPRS
87070603676E7520475052530001	parm - NAME, value="gnu GPRS"
87080603696E7465726E65740001	parm - NAP-ADDRESS, value="internet"
87090689030001	parm - NAP-ADDRTYPE, value=APN
C65A01	characteristic - NAPAUTHINFO
870C069A030001	parm - AUTHTYPE, value=PAP
870D06035A696D62610001	parm - AUTHNAME, value="Zimba"
870E0603536361720001	parm - AUTHSECRET, value="Scar"
01	END (characteristic NAPAUTHINFO)
01	END (characteristic NAPDEF)
C65501	characteristic - NAPDEF
87110603676E755F4353440001	parm - NAPID, value="gnu_CSD"
871006AA030001	parm - BEARER, value=GSM-CSD
87070603676E75204353440001	parm - NAME, value="gnu CSD"
870806032B3535353535353535350001	parm - NAP-ADDRESS, value="+5555555555"
87090687030001	parm - NAP-ADDRESSTYPE, value=E164
870A0690030001	parm - CALLTYPE, value=ANALOG-MODEM

```
87250603393630300001                                    parm - LINKSPEED, value=9600
C65A01                                                  characteristic - NAPAUTHINFO
870C069A030001                                          parm - AUTHTYPE, value=PAP
870D060350756D62610001                                  parm - AUTHNAME, value="Pumba"
870E060353617A750001                                    parm - AUTHSECRET, value="Sazu"
01                                                      END (characteristic NAPAUTHINFO)
01                                                      END (characteristic NAPDEF)
C65501                                                  characteristic - NAPDEF
8711060374696D6F6E5F475052530001                        parm - NAPID, value="timon_GPRS"
871006AB030001                                          parm - BEARER, value="GSM-GPRS"
8707060374696D6F6E20475052530001                        parm - NAME, value="timon GPRS"
87080603696E7465726E65740001                            parm - NAP-ADDRESS, value="internet"
87090689030001                                          parm - NAP-ADDRTYPE, value=APN
C65A01                                                  characteristic - NAPAUTHINFO
870C069A030001                                          parm - AUTHTYPE, value=PAP
870D0603526166696B690001                                parm - AUTHNAME, value="Rafiki"
870E06034B696172610001                                  parm - AUTHSECRET, value="Kiara"
01                                                      END (characteristic NAPAUTHINFO)
01                                                      END (characteristic NAPDEF)
C600015501                                              characteristic - APPLICATION
87360000060377320001                                    parm - APPID, value="w2"
8700013900000603676E752E646B0001                        parm - TO-PROXY, value="gnu.dk"
8707060342726F777365720001                              parm - NAME, value="Browser"
C600015901                                              characteristic - RESOURCE
873A00000603687474703A2F2F7761702E676E752E646B0001 parm - URI, value="http://wap.gnu.dk"
871C01                                                  parm - STARTPAGE
01                                                      END (characteristic RESOURCE)
01                                                      END (characteristic APPLICATION)
C600015501                                              characteristic - APPLICATION
87360000060377340001                                    parm - APPID, value="w4"
8700013900000603746 96D6F6E2E646B0001                   parm - TO-PROXY, value="timon.dk"
8700013400000603687474703A2F2F7761702E74696D6F6E2E646B0001  parm - ADDR, value="http://wap.timon.dk"
01                                                      END (characteristic APPLICATION)
01                                                      END (wap-provisioningdoc)
```

Appendix B. WBXML Binary Example with WSP and WDP Headers

Original XML document:

```
<?xml version="1.0"?>
<!DOCTYPE wap-provisioningdoc PUBLIC "-//WAPFORUM//DTD PROV 1.0//EN"
"http://www.wapforum.org/DTD/prov.dtd">
<wap-provisioningdoc version="1.0">
   <characteristic type="NAPDEF">
      <parm name="NAPID" value="inet"/>
      <parm name="NAME" value="InternetNAPDEF"/>
      <parm name="BEARER" value="GSM-GPRS"/>
      <parm name="NAP-ADDRESS" value="internet"/>
      <parm name="NAP-ADDRTYPE" value="APN"/>
      <parm name="INTERNET"/>
   </characteristic>
   <characteristic type="APPLICATION">
      <parm name="APPID" value="w2"/>
      <parm name="TO-NAPID" value="inet"/>
         <characteristic type="RESOURCE">
            <parm name="URI" value="http://wap.krak.dk"/>
            <parm name="STARTPAGE"/>
         </characteristic>
   </characteristic>
   <characteristic type="BOOTSTRAP">
      <parm name="NAME" value="Sonofon Browser"/>
   </characteristic>
</wap-provisioningdoc>
```

The XML document above converted to binary format (WBXML):

```
030B6A05 696E6574 00C54601 C6550187    ..j. inet ..F. .U..
11068300 01870706 03496E74 65726E65    .... ..... .Int erne
744E4150 44454600 01871006 AB018708    tNAP DEF. .... ....
0603696E 7465726E 65740001 87090689    ..in tern et.. ....
01871401 01C60001 55018736 00000603    .... .... U..6 ....
77320001 87220683 0001C600 01590187    w2.. ."... .... .Y..
3A000006 03687474 703A2F2F 7761702E    :.... .htt p:// wap.
6B72616B 2E646B00 01871C01 0101C656    krak .dk. .... ...V
01870706 03536F6E 6F666F6E 2042726F    .... .Son ofon  Bro
77736572 00010101                      wser ....
```

WBXML explained:

03	Version (1.3)
0B	Public ID : "-//WAPFORUM//DTD PROV 1.0//EN
6A	Encoding : UTF-8
05	String Table - Length
696E657400	String 0: "inet"
C54601	wap-provisioningdoc version=1.0
C65501	characteristic - NAPDEF
871106830001	parm - NAPID, value=String 0 ("inet")
87070603496E7465726E65744E41504445460001	parm - NAME, value="InternetNAPDEF"
871006AB01	parm - BEARER, value=GSM-GPRS
87080603696E7465726E65740001	parm - NAP-ADDRESS, value="internet"
8709068901	parm - NAP-ADDRTYPE, value=APN
871401	parm - INTERNET
01	END (characteristic NAPDEF)
C600015501	characteristic - APPLICATION
87360000060377320001	parm - NAPID, value="w2"
872206830001	parm - TO-NAPID, value=String 0 ("inet")
C600015901	characteristic - RESOURCE
873A00000603687474703A2F2F7761702E6B72616B2E646B0001	parm - URI, value="http://wap.krak.dk"
871C01	parm - INTERNET
01	END - (characteristic RESOURCE)
01	END - (characteristic APPLICATION)
C65601	characteristic - BOOTSTRAP
87070603536F6E6F666F6E2042726F777365720001	parm - NAME, value="Sonofon Browser"
01	END (characteristic BOOTSTRAP)
01	END (wap-provisioningdoc)

WSP Headers

When WBXML is sent to a mobile device it needs to have WSP headers. The headers indicate the MIME type of the content as well as supply various security methods to authenticate the content. The security methods are encoded in the Content-Type header, where the SEC parameter indicates which security method is used, and the MAC parameter holds an HMAC value that is calculated based on the content (the WBXML content encapsulated by the headers) and some other key defined by the SEC parameter.

Calculating the MAC

The following description is taken from [PROVBOOT]:

The MAC is calculated in the following way:
First, the bootstrap document is encoded in the WBXML format [WBXML]. The so-encoded document and the shared secret are then input as the data and key, respectively, for the HMAC calculation [HMAC], based on the SHA-1 algorithm [SHA], as defined in the WTLS specification [WTLS]. The output of the HMAC (M = HMAC-SHA(K, A)) calculation is encoded as a string of hexadecimal digits where each pair of consecutive digits represents a byte. The hexadecimal encoded output from the HMAC calculation is then included in the security information. This calculation is repeated in the ME when checking the validity of the MAC.

In short, this means that the MAC value is the result of using the HMAC algorithm with the WBXML document as A, and a key K determined based on the security method used.

Calculating MAC - USERPIN
When calculating the MAC based on USERPIN, the key K in the above algorithm must be an array of bytes in the range 0x30 - 0x39 (that is, ASCII digits for numbers 0-9). The USERPIN cannot be the empty string.

Calculating MAC - NETWPIN
When calculating the MAC based on NETWPIN, the key K must be the IMSI of the SIM card being provisioned. The IMSI is semi-octet byte encoded before being used as a key.

Calculating MAC - USERNETWPIN
The MAC value calculated using the USERNETWPIN method is a mix between USERPIN and NETWPIN security. The IMSI used in NETWPIN is appended with the user pin used in the USERPIN MAC calculation.

Calculating MAC - USERPINMAC
The USERPINMAC security method is special: the authentication is the same as the USERPIN method, where the last half of the user pin is chosen based on a MAC value that is calculated based on the WBXML content (see PROVBOOT, Section 5.2.1, for a detailed algorithm description).

WBXML example with WSP headers (using USERPIN):

```
01062F1F 2DB69181 92443045 30333443      ../. -... .D0E 034C
30383634 45354537 32444645 41364533      0864 E5E7 2DFE A6E3
42343330 32324133 32423233 39414637      B430 22A3 2B23 9AF7
3600030B 6A05696E 657400C5 4601C6556     ... j.in et.. F..U
01871106 83000187 07060349 6E746572      .... .... ...I nter
6E65744E 41504445 46000187 1006AB01      netN APDE F... ....
87080603 696E7465 726E6574 00018709      .... inte rnet ....
06890187 140101C6 00015501 87360000      .... .... ..U. .6..
06037732 00018722 06830001 C6000159      ..w2 ..." .... ...Y
01873A00 00060368 7474703A 2F2F7761      ..:. ...h ttp: //wa
702E6B72 616B2E64 6B000187 1C010101      p.kr ak.d k... ....
C6560187 07060353 6F6E6F66 6F6E2042      .V.. ...S onof on B
726F7773 65720001 0101                   rows er.. ..
```

WSP headers explained:

```
01                                       TID - [WSP] Ch. 8.2.1
06                                       PDU Type (PUSH) - [WSP] Ch. 8.2.1 + App. A
2F1F2D                                   Headers Length - [WSP] Ch. 8.2.4.1
B6                                       Content-Type - application/vnd.wap.connectivity-wbxml
9181                                     SEC - USERPIN
92                                       MAC
443045303334433038363445345373244464541
3645334234333032324133324232333941463736 MAC value
00                                       End of MAC-value
...                                      WBXML content (see the example)
```

For more information about WSP headers, see [WSP].

WDP Headers

When the WBXML and WSP headers are sent to a mobile device using, for example, Short Message Service (SMS) messages, then Wireless Datagram Protocol (WDP) headers must also be added. WDP headers contain information about, for example, number of message fragments, used sender, and receiver port.

The following shows how WDP headers are added to the above WBXML and WSP headers. The output is fragmented into two messages with a length equal to or less than 140 bytes, making it able to send the output as SMS messages.

SMS 1/2 (140 bytes):

0B05040B 840B8400 03270201 01062F1F	....'.. ../
2DB69181 92443045 30333443 30383634	-... .D0E 034C 0864
45354537 32444645 41364533 42343330	E5E7 2DFE A6E3 B430
32324133 32423233 39414637 3600030B	22A3 2B23 9AF7 6...
6A05696E 657400C5 4601C655 01871106	j.in et.. F..U ....
83000187 07060349 6E746572 6E65744E	....I nter netN
41504445 46000187 1006AB01 87080603	APDE F... ....
696E7465 726E6574 00018709 06890187	inte rnet ....
140101C6 00015501 87360000	.... ..U. .6..

SMS 2/2 (86 bytes):

0B05040B 840B8400 03270202 06037732	....'.. ..w2
00018722 06830001 C6000159 01873A00	..." ....Y ..:.
00060368 7474703A 2F2F7761 702E6B72	...h ttp: //wa p.kr
616B2E64 6B000187 1C010101 C6560187	ak.d k... .... .V..
07060353 6F6E6F66 6F6E2042 726F7773	...Sonof on Brows
65720001 0101	er.. ..

WDP headers in SMS 1/2 explained:

0B	Length of WDP header (11 bytes)
0504	Header info
0B84	Destination port (2948 = WAP Push)
0B84	Source port (2948)
00	Header info
03	Multi-SMS Header Length (SAR)
27	Datagram Reference (must be identical for all the Multi-SMS)
02	Total number of SMS (2)
01	This SMS reference (first SMS)
...	First part of the WSP data described above

WDP headers in SMS 2/2 explained:

0B	Length of header (11 bytes)
0504	Header info
0B84	Destination port (2948 = WAP Push)
0B84	Source port (2948)
00	Header info
03	Multi-SMS Header Length (SAR)
27	Datagram Reference (must be identical for all the Multi-SMS)
02	Total number of SMS (2)
02	This SMS reference (second SMS)
...	Second part of the WSP data described above

For more information about WDP headers, see [WDP].

Appendix C. Frequently Asked Questions

1: How is a provisioning document sent to a Nokia Series 40 mobile device?

An OMA provisioning document is usually sent to a Nokia Series 40 mobile device using one or more SMS messages. The SMS messages contain WDP and WSP headers and the provisioning document, which are encoded using WBXML. The WSP headers are only present in the first SMS message.

Here is an example where three SMS messages are needed to contain the provisioning document:

SMS 1/3

WDP headers	WSP headers	WBXML data	Length = 140 bytes

SMS 2/3

WDP headers	WBXML data		Length = 140 bytes

SMS 3/3

WDP headers	WBXML data		Length <= 140 bytes

2: How should the SMS headers be encoded?
For encoding of SMS headers, which are also called Transfer Protocol Data Units (TPDUs), refer to the [GSM 03.40] and [GSM 03.38] specifications.

Note that messages must be sent as 8-bit messages. The TPDU Octet 11 “TP-Protocol-Identifier” identifying the above layer protocol must be 0xF5 and the TPDU Octet 12 “TP-Data-Coding-Scheme” identifying the data encoding must be 0x15. For further details, see the sections about these TPDUs in the [GSM 03.40] and [GSM 03.38] specifications

3: How should the WDP headers be encoded?
Please refer to the example in Appendix B. For more information about WDP headers, see [WDP].

4: How should the WSP headers be encoded?
Please refer to the example in Appendix B. For more information about WSP header, see [WSP].

5: How should the WBXML be encoded?
Please refer to the example in Appendix B. For general information about WBXML encoding, see [WBXML]. The defined WBXML tokens must be used.

6: Why does nothing happen when an OMA provisioning document is sent to my Nokia Series 40 mobile device?
If the Nokia Series 40 mobile device doesn’t react at all when an OMA provisioning document is being sent to it, then this is probably because one or more of the SMS, WDP, or WSP headers is wrongly encoded. The Nokia Series 40 mobile device will show an error note if the WBXML is defective or if the authentication fails.

7: Why is the error note “Verification failed, settings will be discarded” shown?
This error note appears when the security method NETWPIN is used and the MAC is not as expected. The reason for this could be that a wrong IMSI or wrongly encoded IMSI is used for the MAC calculation. According to [PROVBOOT], Section 6.1, then the "IMSI must be semi-octet representation as defined in [GSM 11.11]," Section 10.3.2. Encoding of, for example, the IMSI "525034370105636" results in hex values "0x59,0x52,0x30,0x34,0x07,0x01,0x65,0x63," which are used as hex input to the MAC calculation. The MAC calculation could also be erroneous.

8: Why is the error note "Failed to save received connection settings" shown?
There can be several reasons for this, for example:

- The Public ID present in the WBXML is not 0x0b. It should be 0x0b, meaning "-//WAPFORUM//DTD PROV 1.0//EN".
- The parameter PROVURL from the BOOTSTRAP element must be unique within the Nokia Series 40 mobile device. If an OMA provisioning document is received with a PROVURL that is already present in the mobile device, then the OMA provisioning document will be rejected.
- The APPLICATION element is mandatory. If an OMA provisioning document is received without this element, then the document is rejected. In some of the early implementations in Nokia Developer Platform 1.0 it was possible to receive provisioning messages without application data. All settings were then saved as browser settings.
- In the Nokia 5100 device it is only possible to save an OMA provisioning message using security method NETWPIN if there have not been any OMA provisioning messages saved earlier using the security method NETWPIN, USERPIN, or USERNETWPIN.
- No security method is used. A security method is mandatory. If an OMA provisioning document is received without a security method, then the document is rejected.
- The received OMA provisioning document is wrongly encoded. Please see Appendix A or Appendix B for examples of correctly encoded documents.

9: How is the MAC calculated?
An example of a third-party tool used to calculate MAC can be found at http://www.slavasoft.com/hashcalc/. The key used for the MAC calculation could, for example, be the string "1234" (0x31,0x32,0x33,0x34), which could represent the user pin used in the security method USERPIN. When using the security method NETWPIN, then the semi-octet encoded IMSI must be used directly, e.g., "0x59,0x52,0x30,0x34,0x07,0x01,0x65,0x63."

10: How can the settings from an OMA provisioning message be edited?
Settings from an OMA provisioning message cannot be edited through the mobile device's UI when they are saved on the Nokia Series 40 mobile device.

11: Can settings for several applications be sent in one OMA provisioning message?
Yes. The idea of OMA provisioning is to have settings for several applications in one provisioning message. If a GSM operator sends a provisioning message with settings for both browser and MMS, then the Nokia Series 40 mobile device can receive and save the settings simultaneously.

The first saved OMA provisioning message is activated by default. Subsequent saved OMA provisioning messages must be manually activated if needed. The assumption is that the first saved OMA provisioning message in Nokia Series 40 mobile devices is from the user's own operator and subsequent ones are from miscellaneous service providers and therefore not activated by default.

12: Can GPRS and CSD settings be saved for one application from one OMA provisioning message?
Currently this is not supported. Two OMA provisioning messages must be sent and saved. The user will then be able to choose between these saved provisioning messages.

Appendix D. Required Files on Smartcard

- EF-DIR (2F00)
  - Must be present for all non 3G cards
  - Must contain a path to the PKCS15 application
- ODF (5031)
  - Must contain a reference to EF(DODF) as specified in the PKCS15 specification
- Tokeninfo (5032)
  - Must contain a version field as specified in the PKCS15 specification
  - Must contain a serialNumber field as specified in the PKCS15 specification
- DODF
  - Must contain a path to the bootstrap file (Data Object) as specified in the PKCS15 specification
- Bootstrap
  - Must contain valid Provisioning content. This can be made by encoding the XML file (containing the Provisioning content) into WBXML. This must be done using tokens from codepage 0 and codepage 1 if defined there. The WBXML file must then be converted from binary to textual format to be written in the bootstrap file.

XML Configuration file

```
<?xml version = "1.0" encoding = "Windows-1252"?>
<CARDSIMDUMP>

<!-- Master File (MF) -->
<DFMF><ID>0</ID><FILEID>3f00</FILEID><DEDICATEDFILES><NUMBER>1</NUMBER><NO0>0</NO0></DEDICATEDFILES><EFTRANS><NUMBER>0</NUMBER></EFTRANS><EF
LIN><NUMBER>1</NUMBER><NO0>0</NO0></EFLIN><MEMLEFT>4f26</MEMLEFT></DFMF>

<!-- EF(Dir) -->
<EFLinear><ID>0</ID><FILEID>2f00</FILEID><PINS><NUMBER>1</NUMBER><NO0>1</NO0></PINS><RECORDS><NUMBER>1</NUMBER><NO0>61304f0ca000000063504b43
532d313551063f007f665f30500199731506019930040402503la00404025032a10404025033</NO0></RECORDS><RECIDX>-1</RECIDX></EFLinear>

<!-- The 7f66 application -->
<DFApps><ID>0</ID><APPID>IDONTKNOW</APPID><FILEID>7f66</FILEID><DEDICATEDFILES><NUMBER>1</NUMBER><NO0>1</NO0></DEDICATEDFILES><EFTRANS><NUMB
ER>0</NUMBER></EFTRANS><EFLIN><NUMBER>0</NUMBER></EFLIN></DFApps>

<!-- The PKCS#15 application -->
<DFApps><ID>1</ID><APPID>a000000063504b43532d3135</APPID><FILEID>5f30</FILEID><DEDICATEDFILES><NUMBER>0</NUMBER></DEDICATEDFILES><EFTRANS><N
UMBER>4</NUMBER><NO0>1</NO0><NO1>2</NO1><NO2>3</NO2><NO3>4</NO3></EFTRANS><EFLIN><NUMBER>0</NUMBER></EFLIN></DFApps>

<!-- TokenInfo file -->
<EFTransparent><ID>1</ID><FILEID>5032</FILEID><PINS><NUMBER>1</NUMBER><NO0>1</NO0></PINS><DATA>3055020100040a980000000000000056f60c0c5363686
c756d626572676572800757494d20312e30030204303018300a0201010605672b010101300a0201020605672b010102a20f300d020200c00201000500030201de</DATA></EF
Transparent>

<!-- EF(ODF) -->
<EFTransparent><ID>2</ID><FILEID>5031</FILEID><PINS><NUMBER>1</NUMBER><NO0>1</NO0></PINS><DATA>a706300404024fcd</DATA></EFTransparent>

<!-- EF(DODF-PROV) -->
<EFTransparent><ID>3</ID><FILEID>4fcd</FILEID><PINS><NUMBER>1</NUMBER><NO0>1</NO0></PINS><DATA>302B30120C09426F6F747374726170030206400401A03
0060604672B0501A10D300B04024fce020100800202b2</DATA></EFTransparent>

<!-- Bootstrap file -->
<EFTransparent><ID>4</ID><FILEID>4fce</FILEID><PINS><NUMBER>1</NUMBER><NO0>1</NO0></PINS><DATA>030B6A23536F6E6F666F6E5F4750525300536F6E6F666
F6E5F50726F787900536F6E6F666F6E00C54503312E310001C60001550187360000060377320001870706831B032042726F777365720001C600015901873A000006036874747
03A2F2F7761702E7476322E646B00018707060354563220686F6D6570616765000187lC0101C60001590187 3A0000060368747470 3A2F2F7761702E6265726C696E67736B652
E646B00018707060342657 26C696E67736B65000101C600015901873A0000060368747470 3A2F2F7761702E6A702E646B0001870706034A796C6C616E647320506F7374656E0
00101C600015901873A0000060368747470 3A2F2F7761702E7961686F6F2E636F6D000187070603596168 6F6F21000101872206830001 01C60001550187360000060377340000
1870706831B03204D4D530001870001390000068 30D0187000134000006036874747 03A2F2F6D6D732E736F6E6F666F6E2E646B000101C60001550187360000060377410001 8
722068300018707060 34D535720494D505300018700013400000603687474703A2F2F6A696D2E64656D6F2E6E646875622E6E65743A31383038302F6373702F6373700001C60
0015701873000000600018D0187310000060361 6C6578616E6465722E70696F7265636B69406E6F6B69612E636F6D000187000132000006033239303100010101C6560187070
6034D7900831B0187180603687474703A2F2F7777772E6D79736F6E6F666F6E2E646B000101C65101871506830D01870706831B0350726F78790001C65201872F06035068797
36963616C50726F787931000187200603323132 2E38382E36342E380001872106850187220683000 1C6530187230603830383000010101 01C6550187110683000187 1006AB0
1870706831B032047505253000187080603696E7465726E6574000187090689018714010101</DATA></EFTransparent>

  </CARDSIMDUMP>
```

Evaluate This Document

In order to improve the quality of documentation, we kindly ask you to fill in the document survey.