*Technical Manual*

# *Browser/OTA Provisioning Whitepaper*

Version 4.0

MOTOROLA
intelligence everywhere™

# Table of Contents

# 1 Introduction

## Summary

Over-the-air provisioning describes the ability to download and install content over a wireless network, usually on demand. This whitepaper will give a detailed explanation of the following:

- Bootstrap process
- Configuration
- User Handset Experience
- OTA Provisioning Security Mechanism
- OTA Provisioning of Browser
- OTA Provisioning of MMS
- OTA Provisioning of SyncML Data Synchronization
- OTA Provisioning of Email
- OTA Provisioning of Other Applications
- Error Conditions
- WAP Provisioning Data Format
- Parameter Mapping Appendix
- OMA Static Conformance Requirements Matrix Appendix
- OTAP Enabled Phones Appendix

For more information on OTAP, consult the OMA Client Provisioning 1.1 specifications: http://www.openmobilealliance.org/release_program/index.html

# References


- Wireless Session Protocol Specification, Open Mobile Alliance, WAP-230-WSP-20010705-a http://www.openmobilealliance.org
- Digital cellular telecommunications system (Phase 2+); Technical realization of the Short Message Service (SMS), Point-to-Point (PP)
- Provisioning Content Version 1.1, Open Mobile Alliance, OMA-WAP-ProvCont-v1_1-20021112-C – http://www.openmobilealliance.org
- Provisioning Bootstrap Version 1.1, Open Mobile Alliance, OMA-WAP-ProvBoot-v1_1-200211120-C – http://www.openmobilealliance.org
- Provisioning User Agent Behavior Version 1.1, Open Mobile Alliance, OMA-WAP-ProvUAB-V1_1-20021113-C - http://www.openmobilealliance.org
- Provisioning Smart Card Specification Version 1.1, Open Mobile Alliance, OMA-WAP-ProvSC-V1_1-20021112-C
- Digital Cellular Telecommunication System (Phase 2+), Specification of the Subscriber Identity Module – Mobile Equipment (SIM-ME) interface (GSM11.11 version 7.2.0 Release 1998)
- Secure Hash Standard, NIST FIPS PUB 180-1, National Institute of Standards and Technology, U.S. Department of Commerce, April 1995.
- HMAC: Keyed-Hashing for Message Authentication, Krawczyk, H., Bellare, M., and Canetti, R., February 1997 – http://www.ietf.org/rfc/rfc2104.txt
- 3rd Generation Partnership Project, Technical Specification Group Core Network; Mobile radio interface layer 3 specification, Core Network Protocols – Stage 3 (Release 4), v 4.5.0 (2001-12)


# Revision History

| Version | Date | Name | Reason |
|---|---|---|---|
| 0.1 | October 20, 2003 | Adam Grabowski | Initial Draft |
| 1.0 | November 3, 2003 | Adam Grabowski | Baseline |
| 1.1 | January 25, 2005 | James Hu | Updated sections for clarity. |
| 2.0 | February 8, 2005 | Doug Michau | Baseline |
| 2.1 | July 5, 2005 | Sreenivasulu Rayanki | Updated with SyncML Data Synchronization |
| 3.0 | July 6, 2005 | Sreenivasulu Rayanki | Baseline |
| 3.1 | July 8, 2005 | Andrey Vostrikov | Updated with Email |

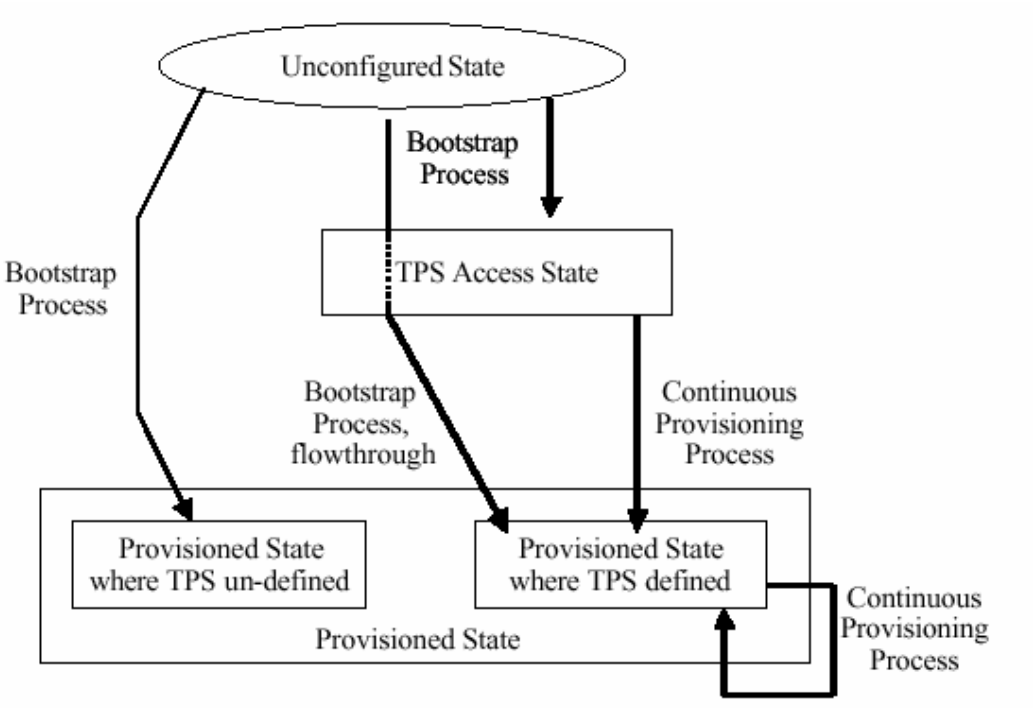
| | | | |
|---|---|---|---|
| 4.0 | August 24, 2005 | Sreenivasulu Rayanki | Baseline |

# Definitions, Abbreviations, Acronyms

| Acronym | Description |
|---|---|
| BCD | Binary Coded Decimal |
| IMSI | International Mobile Subscriber Identity |
| MAC | Message Authentication Code |
| MIB | Motorola Internet Browser |
| MT | Mobile Terminated |
| SIM | Subscriber Information Module |
| OMA | Open Mobile Alliance |
| OTA | Over-the-Air |
| OTAP | Over-the-Air Provisioning |
| SEC | Security |
| TID | Transaction Identifier |
| WBXML | Wireless Binary Extensible Markup Language |
| SyncML DM | SyncML Device Management |
| SyncML DS | SyncML Data Synchronization |
| Web Session | This holds Internet connection settings. Also known as Internet Settings. |

# 2
# Bootstrap Process

The OTAP feature enables carriers and users to provision their handsets for Internet access any time after purchase. The provisioning server will push the data to the handset using the SMS as the bearer. Motorola's implementation of the OTAP feature is based on the OMA Provisioning 1.1 specifications. Please refer to Appendix B for the compliance matrix. The WAP provisioning framework specifies mechanisms to take a terminal from an unconfigured to a fully configured state. Refer to the process diagram below.




Motorola will support the Bootstrap Process to take a handset from the "Unconfigured State" directly to the "Provisioned State where TPS is undefined" state. The Continuous Provisioning Process is not supported. However, Motorola does support alternate means of continuous provisioning such as SyncML DM.

In the Bootstrap Process, it is imperative that security parameters (such as a user pin) are set to prevent unauthorized provisioning documents from configuring the phone.

# 3 Configuration

The carrier may configure many aspects of the OTA Provisioning process. The configuration is setup through handset flexing that is done at the time of manufacture. While the content developer will not have control over these settings, it is important the content developer is aware of them because the configuration will affect the OTAP availability, security, and user flow.

In addition to the carrier configuration, some options can be controlled by the user. These options are discussed in this section too.

## Feature Availability

The following features must be enabled for OTAP support (configured by the carrier).

- Browser Messages Folder – Provisioning messages are stored in this folder.
- Browser Provisioning – Determines whether provisioning messages are processed or discarded.

The following features control other aspects of OTAP

- Whitelist – Additional security (see “Whitelist” section)
- Provision Factory Sessions – Determines whether an OTAP session may update or delete sessions that were provisioned at the time of manufacture. The BOOTSTRAP/NAME in the OTAP session must match the factory session for this to occur – see the section on modification and deletion for more details.
- Disable Default Session – The carrier may set a default session and disable the user from setting any other session as the default.
- Setting Default Session – This setting is only applicable if the user may change the default session. There are three possible settings for this feature:
    1. Off – The OTA provisioning of a new session has no affect on the default browser session.
    2. Automatic – The new session created by OTAP will be set as the default browser session.
    3. Manual – The user will be prompted if she wishes to set the new session created by OTAP as the default browser session.

- Number of Browser Sessions – The maximum limit for browser session is ten, but the carrier may configure fewer sessions.
- SIM Provisioning – If this feature is enabled and a browser session resides on the SIM, then this session will become the new default session. SIM sessions are read-only. So, this feature will not be discussed in this document.

# Whitelist

The whitelist feature provides an additional layer of security. The feature protects against untrusted parties "spamming" the device with provisioning messages. The whitelist is a database that contains up to ten pairs of Originating (TP-OA) / SMSC (MT-SMSC) addresses. Basically, it restricts push (including provisioning) messages to these addresses. A push message that originates from an unauthorized Originating or SMSC Address (not in the whitelist) is discarded.

The carrier may also configure just a SMSC address range check – meaning that only if a push message is sent from a SMSC within a range of addresses will it be accepted.

# User Options

The carrier may configure the handset to allow the user to have some control over the browser messages – includes both the push and provisioning messages.

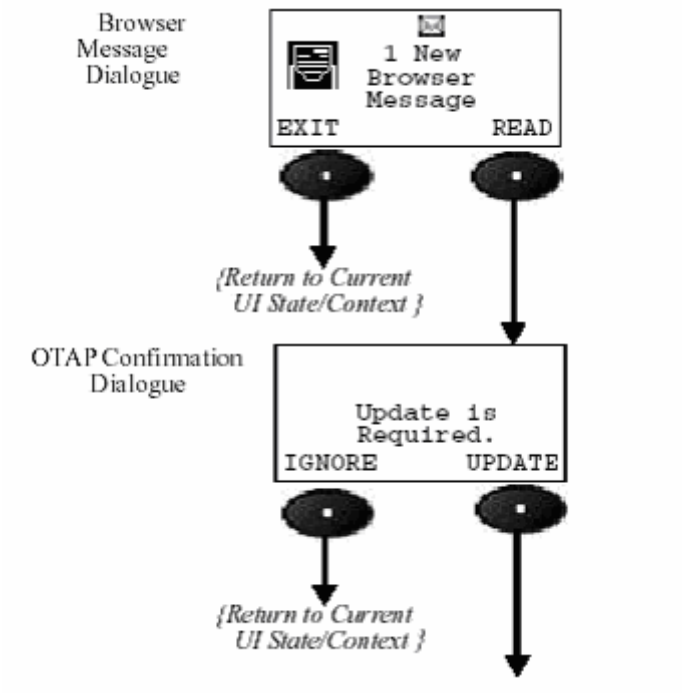
- Browser Message Service – this feature will be available only if the carrier has not turned on the whitelist. This allows the user to screen incoming messages using the following options:
    - Off – all browser messages are discarded
    - Receive All – all browser messages are accepted
    - Restricted – only browser messages that are sent from a specified SMSC are accepted.
- OTAP Session Deletable – the user is able to delete provisioned browser session entries if this feature is set by the carrier.
- MMS Session Editable – the user is able to edit the MMS session entries if this feature is set by the carrier.
- MMS Session Deletable – the user is able to delete the MMS session entries if this feature is set by the carrier.

# Master Clear

A Master Clear operation (under Initial Setup) will delete all of the OTAP and user configured browser and MMS sessions. Only the factory configured sessions (or OTAP updates of the factory sessions) are preserved.

# 4
# User Handset Experience

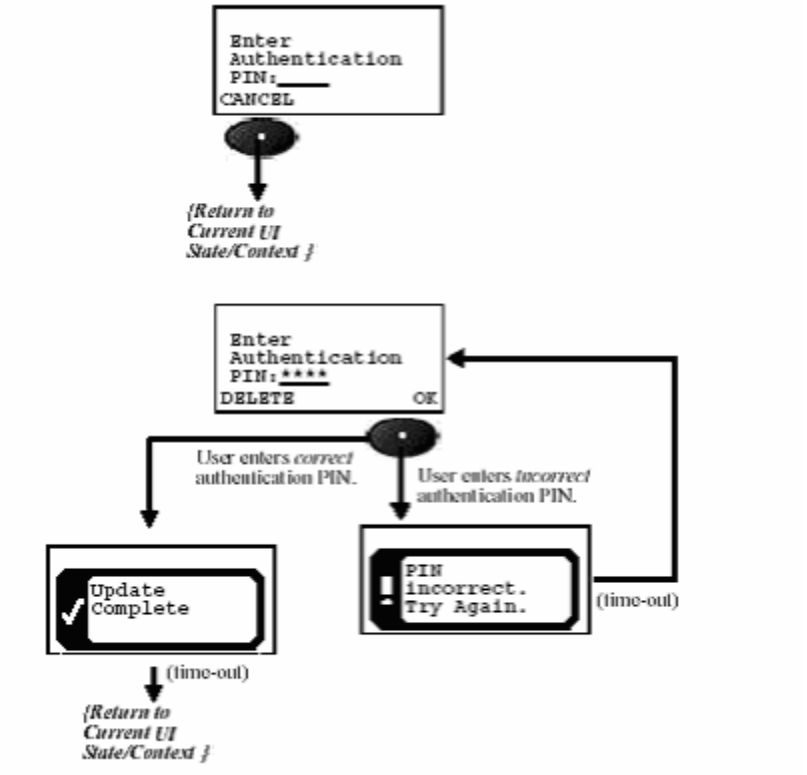
The user will always see the following two screens upon receiving an OTAP message:




The following are the four possible experiences that may exist for the user:

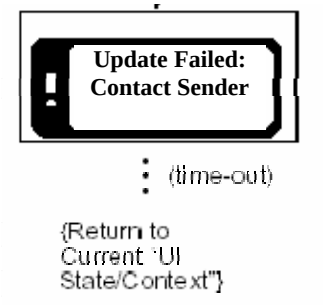
- Valid Digest in case of NETWPIN


- Valid Digest in case of USERPIN or USERNETWPIN

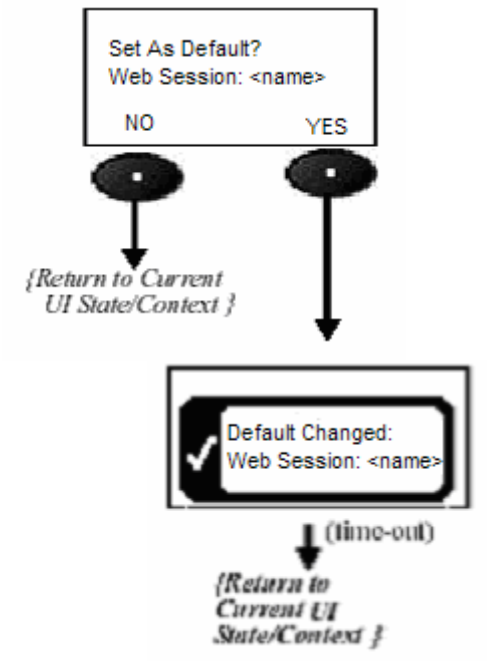



- Invalid Digest


- Unauthenticated Message



An OTAP update may also set the default browser session (as indicated in the "Configuration" section). The possible states for this are off, automatic, and manual. Here is the UI for this feature when it is set to manual:




Please note: Some carriers will also preset the default browser session, and not allow the user to change this. If the handset is configured this way, then OTAP will also not be able to change the default browser session.

# 5
# OTA Provisioning Security Mechanism – Authentication

Whether authentication is required and the method in which it is employed is determined by the gateway sending the provisioning message. This message contains an optional parameter ('sec method') which determines which authentication method is used.

| SEC Value | Authentication Method |
|---|---|
| Not present | No authentication |
| (0) NETWPIN | Authentication Method #1 |
| (1) USERPIN | Authentication Method #2 |
| (2) USERNETWPIN | Authentication Method #2 |
| (3) USERPINMAC | Authentication Method #2 |

The following will be used to define the SEC value for authentication methods:

- USERPIN – Shared secret between the server and client. User enters the value from handset editor prompted upon receiving the provisioning message.
- NETWPIN – Shared secret between the server and client (IMSI). The shared secret is that the IMSI information can be obtained from the SIM card and then subsequently used in calculating a MAC value that is then compared to the one sent along with the provisioning document.
- USERNETWPIN – Shared secret between the server and client. The shared secret is NETWPIN appended with USERPIN.

- USERPINMAC – MAC authentication of document, and USERPIN as shared secret.

The following will be used to define the Authentication Method:

- Authentication Method # 1 – This bootstrap message authentication is performed automatically and does not require user enter any PIN.
- Authentication Method # 2 – This message authentication method requires the mobile handset to create a "Secure Number" Editor prompting the user to enter an authentication PIN (given to the user via some "out-of-band" method).

Important Note: The HMAC value is case-sensitive in existing handsets (fixed in 2H05). The handset will do a case-sensitive comparison between the HMAC given in the provisioning message and the calculated HMAC. The hexadecimal digits A-F must be in UPPER-CASE.

# NETWPIN and USERNETWPIN Authentication

Motorola implementation is based on WAP specification in the Provisioning Bootstrap reference. When NETWPIN or USERNETWPIN is used, the IMSI is used as the network specific shared secret. When this authentication is used as the input to the MAC calculation, the IMSI is on semi-octet representation as defined in the GSM11.11 reference. The following are examples of implementation on a Motorola handset:

1. IMSI contains 15 digits (normally)

For example, assume IMSI number is 310170212226432.

The semi-octet representation (BCD format) will be the following:

0011 ?001 //0x31 or 0x39

0000 0001 // 0x01

0111 0001 // 0x71

0010 0000 // 0x20

0010 0001 // 0x21

0010 0010 // 0x22

0100 0110 // 0x46

0010 0011 // 0x23

Here '?' represents "parity bit" [3GPP 24.008], in this case since there is 15 digits in the IMSI, an odd number, so the parity bit is '1', therefore the first byte is 0x39.

In hex format, it will be 39 01 71 20 21 22 46 23, this will be the value used as the key (together with the WBXML encoded document as the data input) for the HMAC calculation [HMAC], based on the SHA-1 algorithm [SHA], in the case of NETWPIN. The output of the HMAC calculation (20 bytes) is further encoded as a string of 80 hexadecimal digits where each pair of consecutive digits represent a byte, resulting in a 40 bytes MAC value.

In case of USERNETWPIN, for example, the PIN is 1234, the value used as the key for the MAC calculation will be 39 01 71 20 21 22 46 23 31 32 33 34.

2. IMSI contains less than 15 digits (unlikely)

For example, assume IMSI number is 310170212226, it will be padding with FFF. The semi-octet representation (BCD format) will be the following:

```
0011 ?001   //0x31 or 0x39
0000 0001   // 0x01
0111 0001   // 0x71
0010 0000   // 0x20
0010 0001   // 0x21
0010 0010   // 0x22
1111 0110   // 0xF6
1111 1111   // 0xFF
```

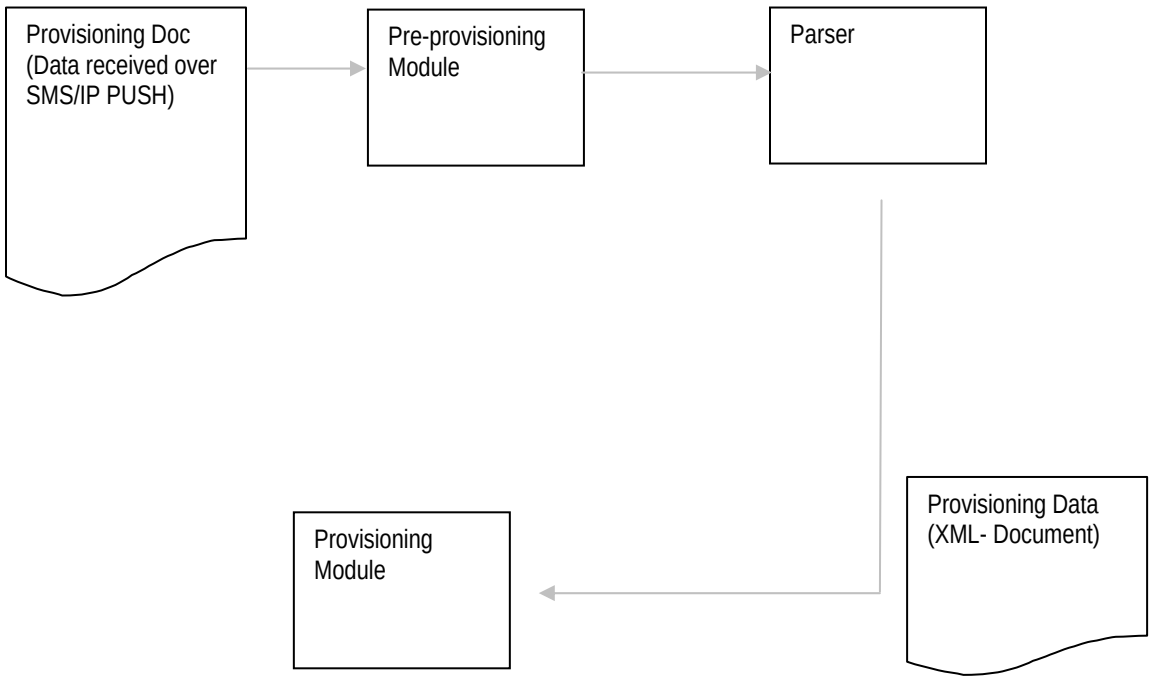
Here '?' represents "parity bit" [3GPP 24.008], in this case since there is 12 digits in the IMSI, an even number, so the parity bit is '0', therefore the first byte is 0x31.

In hex format, it will be 31 01 71 20 21 22 F6 FF, but the last 0xFF is NOT used in the MAC calculation. Only 31 01 71 20 21 22 F6 will be the value used as the key for the MAC calculation in the case of NETWPIN.

In the case of USERNETWPIN, for example, the PIN is 1234, the value used as the key for the MAC calculation will be 31 01 71 20 21 22 F6 31 32 33 34.

# 6
# OTA Provisioning of Browser

The existing architecture (see figure below) requires the parser to parse the received provisioning document into an XML document, allowing the provisioning module to update the corresponding parameters in the phone. The network access point and browser parameters are user-accessible through the “WebSession” or “Internet Profiles” application.




## Code Example

For backwards compatibility, all of the browser parameters are taken from the PXLOGICAL and NAPDEF characteristics. OMA Provisioning 1.1 introduced the APPLICATION characteristic which can also be used to provision the browser parameters. However, the Motorola implementation does not recognize this. Here is an example XML provisioning document:

```
<?xml version="1.0"?>
<!DOCTYPE wap-provisioningdoc PUBLIC "-//WAPFORUM//DTD PROV 1.0//EN"
"http://www.wapforum.org/DTD/prov.dtd">
<wap-provisioningdoc>
<characteristic type="BOOTSTRAP">
      <parm name="PROXY-ID" value="sdp.xyz.com"/>
      <parm name="NAME" value="xyz"/>
</characteristic>
<characteristic type="PXLOGICAL">
      <parm name="PROXY-ID" value="sdp.xyz.com"/>
      <parm name="NAME" value="MOT BROWSING"/>
      <parm name="STARTPAGE" value="http://www.web.com"/>
      <characteristic type="PXPHYSICAL">
            <parm name="PHYSICAL-PROXY-ID" value="Gateway 1"/>
            <parm name="PXADDR" value="217.171.129.2"/>
            <parm name="PXADDRTYPE" value="IPV4"/>
            <parm name="TO-NAPID" value="NAP 1"/>
            <characteristic type="PORT">
                  <parm name="PORTNBR" value="8799"/>
                  <parm name="SERVICE" value="OTA-HTTP-PO"/>
            </characteristic>
      </characteristic>
</characteristic>
<characteristic type="NAPDEF">
      <parm name="NAPID" value="NAP 1"/>
      <parm name="BEARER" value="GSM-GPRS"/>
      <parm name="NAME" value="XYZ GPRS"/>
      <parm name="NAP-ADDRESS" value="xyz.co.uk"/>
      <parm name="NAP-ADDRTYPE" value="APN"/>
</characteristic>
</wap-provisioningdoc>
```

The following describes the values parsed by the client with regards to the example above (Please refer to Appendix A for the complete mapping of parameters):

- Create a browser session with the following parameters:
  - Name: “MOT BROWSING”
  - Homepage: “http://www.web.com”
  - Service Type 1: “HTTP”
  - Gateway IP1: “217.171.129.2”
  - Port 1: “8799”
  - APN: “xyz.co.uk”
  - *All other non-specified parameters will have a default value or left blank.*

# Operating Constraints

The following is a list of operating implementation constraints:

- No two sessions can have the same BOOTSTRAP/NAME. If an entry exists with the same BOOTSTRAP/NAME, it will be updated.

- The OTA provisioned session is read only (i.e. the user can not edit). To edit the content, the user must make a copy and edit the copied version.
- A maximum of 10 total browser sessions (unless configured for less) can be stored at anytime in the phone (regardless if set at time of manufacture, OTA provisioned, or user created).
- BOOTSTRAP
    a. The BOOTSTRAP characteristic is optional. However, this should be used if there is a desire to update or delete this session at a later time (see later section)
    b. If the BOOTSTRAP is used, then the PROXY-ID must match the PXLOGICAL PROXY-ID.
    c. If the BOOTSTRAP is missing (or invalid), the session will still be stored with a blank BOOTSTRAP name.
    d. Important Note: Another provisioning document without a BOOTSTRAP characteristic would overwrite this session since both this document and the new document have a blank BOOTSTRAP name. Essentially, there can be only one session in the database with a blank BOOTSTRAP name.
- PXLOGICAL
    a. The NAME is used to identify the session.
    b. The STARTPAGE is used as the browser homepage
    c. Motorola supports up to two PXPHYSICAL characteristics per PXLOGICAL (proxy 1 and proxy 2).
- PXADDRTYPE – Starting in 2H05, Motorola handsets will support a domain name for the proxy (i.e., “ALPHA”).
- NAPDEF
    a. Motorola supports up to three (1 GPRS and 2 CSD) NAPDEF characteristics. ALL NAPDEF characteristics must be referenced by each PXPHYSICAL characteristic. Any unreferenced NAPDEF will be ignored.
    b. Only the AUTHNAME and AUTHSECRET from the NAPAUTHINFO characteristic are used. The authentication type (PAP or CHAP) is determined by the carrier and is configured at the time of manufacture.
- PORT
    a. The SERVICE parameter must be included.
    b. The possible values are listed below with the most common ports (any port can be used):
        i. CL-WSP: WAP connection-less (e.g., port 9200)
        ii. CO-WSP: WAP connection-oriented (e.g., port 9201)
        iii. CL-SEC-WSP: WAP secure connection-less (e.g., port 9202)
        iv. CO-SEC-WSP: WAP secure connection-oriented (e.g., port 9203)
        v. OTA-HTTP-PO: HTTP/WAP 2.0 (e.g., port 80 or 8080)

vi. OTA-HTTP-TLS-PO: Secure HTTP/WAP 2.0 (e.g., port 443)

- The user is not allowed to create a browser session with a name identical to an OTA provisioned session. An error message will be displayed if this is attempted.
- If the handset receives an OTAP session with a PXLOGICAL/NAME that matches an existing PXLOGICAL/NAME (or the user session name), then the new session name will be appended with a number "1…9" to identify it uniquely. If the name is already at max length, the last letter will be cut off in order to append the number "1...9".

# Modification/Deletion of Provisioned Session by Operator

To modify or delete a session by service provider, the OTAP provider needs to send a session with a BOOTSTRAP/NAME that matches an existing sessions BOOTSTRAP/NAME.

The modification is actually a simple two part operation: delete the existing session and add the new session. A modification example is not given because it is identical to the browser provisioning example.

To delete a session, the provisioning document must only contain the BOOTSTRAP characteristic. The following is an example of an operator deleting the session provisioned previously according to the document above.

```
<?xml version="1.0"?>
<!DOCTYPE wap-provisioningdoc PUBLIC "-//WAPFORUM//DTD PROV 1.0//EN"
"http://www.wapforum.org/DTD/prov.dtd">
<wap-provisioningdoc>
<characteristic type="BOOTSTRAP">
      <parm name="NAME" value="xyz"/>
</characteristic>
</wap-provisioningdoc>
```

If this deletion is successful, the phone will display “Update Complete.” If the deletion fails for any reason (e.g. the session does not exist), the phone will display “Update Failed.”

Important Notes

- If the original provisioning document did not include the BOOTSRAP characteristic, the operator will be able to delete the existing session by sending a provisioning document containing only the BOOTSTRAP characteristic with the NAME parameter set to value “” (empty string).
- An OTAP session can only modify/delete other sessions created through the OTAP process. They cannot modify/delete user-created sessions. The OTAP may be able to modify/delete the sessions that were provisioned at the time of manufacture (see “Configuration”).

# 7
# OTA Provisioning of MMS

In order to extend provisioning support to MMS, Motorola will support the “APPLICATION” Characteristic in the provisioning document as well as the specific parameters from this characteristic.

## Code Example

Below is an example of a XML document for configuring the MMS Specific parameters. This example only highlights the extra parameters in the XML document for MMS. These parameters must be combined with the browser parameters (PXLOGICAL and NAPDEF) for a full provisioning document.

The application identifier for MMS is “w4.” The application id is defined in the OMNA Push Application ID document located at http://www.openmobilealliance.org/tech/omna/omna-push-app-id.htm.

```
<characteristic type="APPLICATION">
      <parm name="APPID" value="w4"/>
      <parm name="NAME" value="MOT MMS"/>
      <parm name="ADDR" value="http://mms.um.xyz.co.uk:10021/mmsc"/>
</characteristic>
```

The following describes the values parsed by the client with regards to the example above:

- Create a browser session with name “MOT MMS” (assuming that the PXLOGICAL had the name “MOT MMS”).
- Create a MMS session with the following parameters:
    - Service Name: “MOT MMS”
    - Server Name: “http://mms.um.xyz.co.uk:10021/mmsc”
    - WebSession Name: “MOT MMS”

## Operating Constraints

The following is a list of operating implementation constraints:

- The Motorola client will always use the same name (as received in the application characteristics) for MMSC and the corresponding browser session entry. This way, Motorola can extend support to multiple servers and tie them to corresponding connection parameters.
- The NAME must match with the PXLOGICAL NAME.
- If the NAME does not match, then the entire APPLICATION characteristic will be discarded.
- All other parameters in the APPLICATION characteristic are ignored.
- A maximum of 3 MMS sessions can be stored at anytime in the phone (regardless if set at time of manufacture, OTA provisioned, or user created).
- The last provisioned entry will be the selected as the default MMSC.
- If a browser session is deleted by the user or by OTAP, then any MMS that references the deleted browser session will also be deleted.
- If the user deletes a MMS session, then this will have no affect on the browser sessions since the session can be used by other applications (e.g., GPRS “Always-on” feature, Browser, or KJava Networking).

# 8
# OTA Provisioning of SyncML Data Synchronization

In order to extend provisioning support to SyncML Data Synchronization, Motorola will support the “APPLICATION” Characteristic in the provisioning document as well as the specific parameters and sub-characteristics from this characteristic.

## Code Example

Below is an example of a XML document for configuring the SyncML DS Specific parameters.

The application identifier for SyncML DS is “w5.” The application id is defined in the OMNA Push Application ID document located at http://www.openmobilealliance.org/tech/omna/omna-push-app-id.htm.

```
<?xml version="1.0"?>
<!DOCTYPE wap-provisioningdoc PUBLIC "-//WAPFORUM//DTD PROV
1.0//EN" "http://www.wapforum.org/DTD/prov.dtd">
<wap-provisioningdoc>
  <characteristic type=”APPLICATION”>
    <parm name=”APPID” value=”w5”/>
    <parm name=”NAME” value=”DataSync”/>
    <parm name=”ADDR” value=”www.datasync.org/servlet/syncit”/>
    <characteristic type=”APPAUTH”>
      <parm name=”AAUTHLEVEL” value=”APPSRV”/>
      <parm name=”AAUTHTYPE” value=”HTTP-BASIC”/>
      <parm name=”AAUTHNAME” value=”William”/>
      <parm name=”AAUTHSECRET” value=”will123”/>
    </characteristic>
    <characteristic type=”RESOURCE”>
      <parm name=”URI” value=”./addressbook/myaddresses”/>
      <parm name=”AACCEPT” value=”text/x-vcard”/>
    </characteristic>
    <characteristic type=”RESOURCE”>
      <parm name=”URI” value=”./appointments/myappointments”/>
      <parm name=”AACCEPT” value=”text/x-vcalendar”/>
    </characteristic>
```

```
        </characteristic>
</wap-provisioningdoc>
```

The following describes the values parsed by the client with regards to the example above:

- Create a SyncML DS Server session with the following parameters:
    - Name: “DataSync”
    - URL: “www.datasync.org/servlet/syncit”
    - User Name: “William”
    - Password: “will123”
    - Data Paths:
        - Phonebook: “./addressbook/myaddresses”
        - Datebook: “./appointments/myappointments”

# Operating Constraints

The following is a list of operating implementation constraints:

- Motorola supports provisioning of up to three SyncML DS Server sessions. Each session can have up to two resource data paths (Phonebook & Datebook).
- The SyncML DS Server session will use the default browser session to establish data connection. Therefore, Motorola will not support the Proxy and NAP for the SyncML DS Provisioning. Also, modifying the default browser session or adding a new default browser session will affect the data parameters for this application too.
- Resource data paths: Mailing List & To-Dos can not be provisioned.
- APPAUTH
    a. AAUTHLEVEL - Motorola supports only application level authentication (i.e., APPSRV). If OTAP message contains any other methods, the message would be discarded.
    b. AAUTHTYPE – Motorola supports only basic authentication method (i.e., HTTP-BASIC). If OTAP message contains any other methods, the message would be discarded.
- All other parameters in the APPLICATION characteristic are ignored.
- To update an already provisioned SyncML DS session, one needs to send the new OTAP document with the same Name as an existing session.

# 9
# OTA Provisioning of Email

The Email application creates/updates account(s) and web session(s) using provisioned data. Account's data is provisioned in two paired APPLICATION characteristics: one of them contains data for send operations, other– for receiving operations. Each account in the provisioning document should contain related NAPDEF characteristic to configure web session. This web session will be used by Email to establish connection and perform operations for given account.

## Code Example

Below is an example of a XML document for configuring the Email account and related Web Session.

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

      <characteristic type="APPLICATION">
          <!-- POP3 settings -->
          <parm name="APPID" value="110"/>
          <parm name="NAME" value="account_one"/>
          <parm name="PROVIDER-ID" value="1"/>
          <parm name="TO-NAPID" value="NAP 1"/>
          <characteristic type="APPADDR">
            <parm name="ADDR" value="pop.mailserver.com"/>
          </characteristic>
          <characteristic type="APPAUTH">
            <parm name="AAUTHNAME" value="user_id"/>
            <parm name="AAUTHSECRET" value="password"/>
          </characteristic>
        </characteristic>

        <characteristic type="APPLICATION">
          <!-- SMTP settings -->
          <parm name="APPID" value="25"/>
          <parm name="PROVIDER-ID" value="1"/>
          <parm name="FROM" value="account_one@mailserver.com"/>
```

```
          <characteristic type="APPADDR">
            <parm name="ADDR" value="smtp.mailserver.com"/>
          </characteristic>
        </characteristic>

        <characteristic type="NAPDEF">
          <!-- Web Session /Internet Settings -->
          <parm name="NAPID" value="NAP 1"/>
          <parm name="NAME" value="Email_GPRS"/>
          <parm name="NAP-ADDRESS" value="internet.provider.com"/>
          <parm name="NAP-ADDRTYPE" value="APN"/>
          <characteristic type="NAPAUTHINFO">
            <parm name="AUTHNAME" value="user"/>
            <parm name="AUTHSECRET" value="pswd"/>
          </characteristic>
        </characteristic>

</wap-provisioningdoc>
```

In the above example, the Email account (POP3/SMTP) with following settings will be created:

- Account Name: “account_one”
- User ID: “user_id”
- Password: “password”
- Return address: “account_one@mailserver.com”
- Web Session: “Email_GPRS”
- Protocol: POP3
- Sending host: “smtp.mailserver.com”
- Sending port: 25
- Receiving host: “pop.mailserver.com”
- Receiving port: 110

And Web Session:

- Name: “Email_GPRS”
- GPRS APN: “internet.provider.com”
- User Name: “user”
- Password: “pswd”

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

      <characteristic type="APPLICATION">
          <parm name="APPID" value="143"/>
```

```
          <!-- IMAP4 settings -->
          <parm name="NAME" value="account_two"/>
          <parm name="PROVIDER-ID" value="1"/>
          <parm name="TO-NAPID" value="NAP 1"/>
          <characteristic type="APPADDR">
            <parm name="ADDR" value="imap.mailserver.com"/>
          </characteristic>
          <characteristic type="APPAUTH">
            <parm name="AAUTHNAME" value="user_id"/>
            <parm name="AAUTHSECRET" value="password"/>
          </characteristic>
        </characteristic>

        <characteristic type="APPLICATION">
          <!-- SMTP settings -->
          <parm name="APPID" value="25"/>
          <parm name="PROVIDER-ID" value="1"/>
          <parm name="FROM" value="account_two@mailserver.com"/>
          <characteristic type="PORT">
            <parm name="PORTNBR" value="25000"/>
          </characteristic>
          <characteristic type="APPADDR">
            <parm name="ADDR" value="smtp.mailserver.com"/>
          </characteristic>
        </characteristic>

        <characteristic type="NAPDEF">
          <!-- Web Session /Internet Settings -->
          <parm name="NAPID" value="NAP 1"/>
          <parm name="NAME" value=" Email_CSD"/>
          <parm name="NAP-ADDRESS" value="+79021234567"/>
          <parm name="NAP-ADDRTYPE" value="E164"/>
          <characteristic type="NAPAUTHINFO">
            <parm name="AUTHNAME" value="csd_user"/>
            <parm name="AUTHSECRET" value="csd_pswd"/>
          </characteristic>
        </characteristic>
</wap-provisioningdoc>
```

In the above example, the Email account (IMAP4/SMTP) with following settings will be created:

- Account Name: “account_two”
- User ID: “user_id”
- Password: “password”
- Return address: “account_two@mailserver.com”
- Web Session: “Email_CSD”
- Protocol: IMAP4
- Receiving host: “imap.mailserver.com”
- Receiving port: 143
- Sending host: “smtp. mailserver.com”

- Sending port: 25000 (non standard port)

And Web Session:

- Name: “Email_CSD”
- CSD No. 1: “+79021234567”
- User Name 1: “csd_user”
- Password 1: “csd_pswd”

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

      <characteristic type="APPLICATION">
          <!-- POP3 settings -->
          <parm name="APPID" value="110"/>
          <parm name="NAME" value="account_one"/>
          <parm name="PROVIDER-ID" value="1"/>
          <parm name="TO-NAPID" value="NAP 1"/>
          <characteristic type="APPADDR">
            <parm name="ADDR" value="pop.mailserver.com"/>
            <characteristic type="PORT">
              <parm name="PORTNBR" value="995"/>
              <parm name="SERVICE" value="995"/>
            </characteristic>
          </characteristic>
          <characteristic type="APPAUTH">
            <parm name="AAUTHNAME" value="user_id"/>
            <parm name="AAUTHSECRET" value="password"/>
          </characteristic>
        </characteristic>

        <characteristic type="APPLICATION">
          <!-- SMTP settings -->
          <parm name="APPID" value="25"/>
          <parm name="PROVIDER-ID" value="1"/>
          <parm name="FROM" value="account_one@mailserver.com"/>
          <characteristic type="APPADDR">
            <parm name="ADDR" value="smtp.mailserver.com"/>
            <characteristic type="PORT">
              <parm name="PORTNBR" value="25"/>
              <parm name="SERVICE" value="STARTTLS"/>
            </characteristic>
          </characteristic>
        </characteristic>

        <characteristic type="NAPDEF">
          <!-- Web Session /Internet Settings -->
          <parm name="NAPID" value="NAP 1"/>
          <parm name="NAME" value="Email_GPRS"/>
          <parm name="NAP-ADDRESS" value="internet.provider.com"/>
          <parm name="NAP-ADDRTYPE" value="APN"/>
          <characteristic type="NAPAUTHINFO">
            <parm name="AUTHNAME" value="user"/>
```

```
            <parm name="AUTHSECRET" value="pswd"/>
          </characteristic>
        </characteristic>

</wap-provisioningdoc>
```

In the above example, the Email account (SSL connection) with following settings will be created:

- Account Name: “account_one”
- User ID: “user_id”
- Password: “password”
- Return address: “account_one@mailserver.com”
- Web Session: “Email_GPRS”
- Protocol: POP3
- Sending host: “smtp.mailserver.com”
- Sending port: 25
- Receiving host: “pop.mailserver.com”
- Receiving port: 995
- Security->Use SSL for sending: yes
- Security->Use SSL for receiving: yes

And Web Session:

- Name: “Email_GPRS”
- GPRS APN: “internet.provider.com”
- User Name: “user”
- Password: “pswd”

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

      <characteristic type="APPLICATION">
          <!-- POP3 settings -->
          <parm name="APPID" value="110"/>
          <parm name="NAME" value="account_one"/>
          <parm name="PROVIDER-ID" value="1"/>
          <parm name="TO-NAPID" value="NAP 1"/>
          <characteristic type="APPADDR">
            <parm name="ADDR" value="pop.mailserver.com"/>
          </characteristic>
          <characteristic type="APPAUTH">
            <parm name="AAUTHNAME" value="receiving_user_id"/>
            <parm name="AAUTHSECRET" value=" receiving_password"/>
          </characteristic>
```

```
        </characteristic>

        <characteristic type="APPLICATION">
          <!-- SMTP settings -->
          <parm name="APPID" value="25"/>
          <parm name="PROVIDER-ID" value="1"/>
          <parm name="FROM" value="account_one@mailserver.com"/>
          <characteristic type="APPADDR">
            <parm name="ADDR" value="smtp.mailserver.com"/>
          </characteristic>
          <characteristic type="APPAUTH">
            <parm name="AAUTHNAME" value="sending_user_id"/>
            <parm name="AAUTHSECRET" value=" sending_password"/>
          </characteristic>
        </characteristic>

        <characteristic type="NAPDEF">
          <!-- Web Session /Internet Settings -->
          <parm name="NAPID" value="NAP 1"/>
          <parm name="NAME" value="Email_GPRS"/>
          <parm name="NAP-ADDRESS" value="internet.provider.com"/>
          <parm name="NAP-ADDRTYPE" value="APN"/>
          <characteristic type="NAPAUTHINFO">
            <parm name="AUTHNAME" value="user"/>
            <parm name="AUTHSECRET" value="pswd"/>
          </characteristic>
        </characteristic>

</wap-provisioningdoc>
```

In the above example, the Email account (separate credentials for both sending and receiving) with following settings will be created:

- Account Name: “account_one”
- Receiving User ID: “receiving_user_id”
- Receiving Password: “receiving_password”
- Sending User ID: “sending_user_id”
- Sending Password: “sending_password”
- Return address: “account_one@mailserver.com”
- Web Session: “Email_GPRS”
- Protocol: POP3
- Sending host: “smtp.mailserver.com”
- Sending port: 25
- Receiving host: “pop.mailserver.com”
- Receiving port: 110

And Web Session:

- Name: “Email_GPRS”

- GPRS APN: “internet.provider.com”
- User Name: “user”
- Password: “pswd”

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

      <characteristic type="APPLICATION">
          <!-- POP3 settings -->
          <parm name="APPID" value="110"/>
          <parm name="NAME" value="account_one"/>
          <parm name="PROVIDER-ID" value="1"/>
          <parm name="TO-NAPID" value="NAP 1"/>
          <characteristic type="APPADDR">
            <parm name="ADDR" value="pop.mailserver.com"/>
          </characteristic>
          <characteristic type="APPAUTH">
            <parm name="AAUTHNAME" value="user_id"/>
            <parm name="AAUTHSECRET" value="password"/>
          </characteristic>
        </characteristic>

        <characteristic type="APPLICATION">
          <!-- SMTP settings -->
          <parm name="APPID" value="25"/>
          <parm name="PROVIDER-ID" value="1"/>
          <parm name="FROM" value="account_one@mailserver.com"/>
          <characteristic type="APPADDR">
            <parm name="ADDR" value="smtp.mailserver.com"/>
          </characteristic>
        </characteristic>

      <characteristic type="APPLICATION">
          <!-- IMAP4 settings -->
          <parm name="APPID" value="143"/>
          <parm name="NAME" value="account_two"/>
          <parm name="PROVIDER-ID" value="2"/>
          <parm name="TO-NAPID" value="NAP 1"/>
          <characteristic type="APPADDR">
            <parm name="ADDR" value="imap.mailserver.com"/>
          </characteristic>
          <characteristic type="APPAUTH">
            <parm name="AAUTHNAME" value="user_id"/>
            <parm name="AAUTHSECRET" value="password"/>
          </characteristic>
        </characteristic>

        <characteristic type="APPLICATION">
          <!-- SMTP settings -->
          <parm name="APPID" value="25"/>
          <parm name="PROVIDER-ID" value="2"/>
          <parm name="FROM" value="account_two@mailserver.com"/>
```

```
          <characteristic type="APPADDR">
            <parm name="ADDR" value="smtp.mailserver.com"/>
          </characteristic>
        </characteristic>

         <characteristic type="NAPDEF">
          <!-- Web Session /Internet Settings -->
          <parm name="NAPID" value="NAP 1"/>
          <parm name="NAME" value="Email_GPRS"/>
          <parm name="NAP-ADDRESS" value="internet.provider.com"/>
          <parm name="NAP-ADDRTYPE" value="APN"/>
          <characteristic type="NAPAUTHINFO">
            <parm name="AUTHNAME" value="user"/>
            <parm name="AUTHSECRET" value="pswd"/>
          </characteristic>
        </characteristic>

</wap-provisioningdoc>
```

In the above example, the Email account (multiple accounts) with following settings will be created:

- Account Name: “account_one”
- User ID: “user_id”
- Password: “password”
- Return address: “account_one@mailserver.com”
- Web Session: “Email_GPRS”
- Protocol: POP3
- Sending host: “smtp.mailserver.com”
- Sending port: 25
- Receiving host: “pop.mailserver.com”
- Receiving port: 110

- Account Name: “account_two”
- User ID: “user_id”
- Password: “password”
- Return address: “account_two@mailserver.com”
- Web Session: “Email_GPRS”
- Protocol: IMAP4
- Receiving host: “imap.mailserver.com”
- Receiving port: 143
- Sending host: “smtp. mailserver.com”
- Sending port: 25

And Web Session:

- Name: “Email_GPRS”
- GPRS APN: “internet.provider.com”
- User Name: “user”
- Password: “pswd”

# Operating Constraints

The following is a list of operating implementation constraints:

- The PROVIDER-ID must match between APPLICATION characteristics to bind settings for POP3 (or IMAP4) and SMTP together.
- TO-NAPID and NAME parameters must be supplied in POP3/IMAP4 APPLICATION characteristic. These parameters are optional in SMTP characteristic.
- If TO-NAPID and NAME parameters are provided in both APPLICATON characteristics (POP3/IMAP4 and SMTP) with the same PROVIDER-ID, then it must match, otherwise account settings will be discarded as invalid.
- If NAME parameter is absent in both POP3/IMAP4 and SMTP characteristics, a new account will be created with auto-generated name. Updating of such accounts is impossible through OTAP.
- If “User ID” and “Password” parameters are not provisioned, a new account will be created with empty values. But this account can not be used in Email operations until User fills in “User ID” and “Password” fields manually.
- Email uses PORT/SERVICE and PORT/PORTNBR parameters combination to set secured connection to mail servers. If PORT/SERVICE is set to STARTTLS, PORT/PORTNBR must not contain dedicated port number for POP3/IMAP4 protocols (995/993 respectively), SMTP protocol has no restrictions. If PORT/SERVICE is set to dedicated port number (995/993 for POP3/IMAP4 respectively) PORT/ PORTNBR must contain the same value.
  Valid combinations are:

| APPLICATION/APPID | PORT/PORTNBR | PORT/SERVICE |
|---|---|---|
| 110 | 995 | 995 |
| 110 | Any, except 995 | STARTTLS |
| 143 | 993 | 993 |
| 143 | Any, except 993 | STARTTLS |
| 25 | Any | STARTTLS |

  To setup unsecured connection on non default port number, setup PORT/ PORTNBR to specific value and PORT/SERVICE parameter must be omitted.
- One OTAP document can contain settings for several accounts and web sessions.
- If document contains more than two APPLICATION characteristics with same PROVIDER-ID, only the first occurrence of POP3/IMAP4 and SMTP characteristics will be used to bind them together. In the below example second and fourth characteristics will be ignored.

```
…
        <characteristic type="APPLICATION">
          <parm name="APPID" value="110"/>
          <parm name="PROVIDER-ID" value="1"/>
          …
        </characteristic>

        <characteristic type="APPLICATION">
          <parm name="APPID" value="143"/>
          <parm name="PROVIDER-ID" value="1"/>
          …
        </characteristic>

        <characteristic type="APPLICATION">
          <parm name="APPID" value="25"/>
          <parm name="PROVIDER-ID" value="1"/>
          <parm name="FROM" value="account@server_one.com"/>
          …
        </characteristic>

        <characteristic type="APPLICATION">
          <parm name="APPID" value="25"/>
          <parm name="PROVIDER-ID" value="1"/>
          <parm name="FROM" value="account@server_two.com"/>
          …
        </characteristic>
…
```

- Not all parameters in Email Account settings can be provisioned by OTAP. The one which can be provisioned is listed in "Error! Reference source not found.".
- Not all parameters of Web session are used by Email, so only related parameters from NAPDEF are stored in Web session. See "Error! Reference source not found." for Web Session parameters supported by Email.
- To update the e-mail account, OTAP provider needs to send an account with the APPLICATION/NAME value matching an existing account's name.
- To update the Web Session, OTAP provider needs to send the Web Session with the NAPDEF/NAME value matching an existing Web Session's name.
- Email accounts can not be deleted through OTAP.

# 10 OTA Provisioning of Other Applications

As Motorola extends the client provisioning support for other applications, newer versions of this document will be published on the MotoCoder support site. Other applications use the connectivity parameters in the browser sessions to establish a data connection. Since the browser sessions may be added/updated/deleted through OTAP, these applications will also indirectly be affected.

## KJava

KJava Network Applications will use a session named “Java Session.” If this session does not exist, then they will use the default browser session. To provision or update the KJava network connectivity parameters, set the PXLOGICAL/NAME to “Java Session” in the provisioning message.

## “Always-On”

“Always-On” application will use the default browser session to establish a data connection. Therefore, modifying the default browser session or adding a new default browser session will affect the data parameters for this application too.

## Streaming

The streaming application works similar to the KJava Network applications when choosing a browser session to establish a data connection. The application will first choose a particular browser session (the name is configured by the carrier). If this session does not exist, then it will use the parameters in the default browser session to establish a data connection.

# 11
# Error Conditions

The user will always be notified if any error occurs during provisioning. The following is a list of possible error conditions:

- The provisioning document is incomplete or erroneous.
- The delete operation fails (e.g. the session has been deleted by the user before).
- Authentication fails.
- Database full - either browser or MMS sessions are full.

## Common Issues

Please use this checklist if the handset does not display the new provisioning message:

- Is the feature is enabled (both the browser message inbox and provisioning)?
- Are the Originating Address and SMSC authorized (via Whitelist or User Restriction)?

The handset displays the message, but the update fails. Please use this checklist:

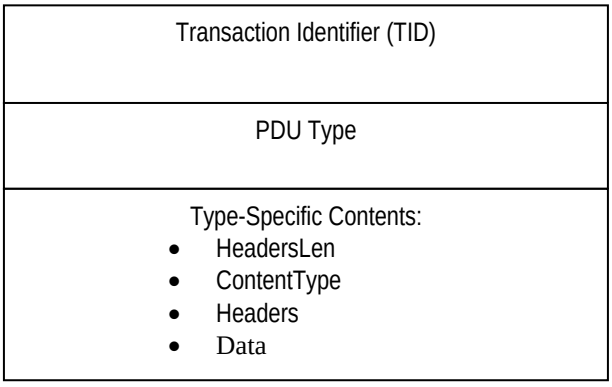
- Is there space available to add the new session?
- Is a supported security mechanism used (NETWPIN, USERPIN, or USERNETWPIN)?
- Are the hexadecimal digits for the HASH value in all capital letters?
- Is the OTAP session modifying or adding a new session (i.e., does the OTAP message have a unique BOOTSTRAP/NAME)?
- Does the BOOTSTRAP/PROXY-ID match the PXLOGICAL/PROXY-ID?

The provisioning is successful, but the browser or MMS is not connecting. Please use this checklist:

- Double-check all of the parameters in the phone for accuracy.
- Make sure that a PORT/SERVICE is specified.
- The minimal parameters for a browser session are the name, startpage, service type, port, and the GPRS APN.

# 12
# WAP Provisioning Data Format

This section describes WAP provisioning information sent using the WAP Push protocol over the GSM SMS bearer. WAP WSP defines the generic Protocol Data Unit (PDU) shown below:




- TID: uint8 will be present for connectionless WSP PDUs according to the WAP WSP specification. The value of 0x01 will be used in this example.
- PDU Type: uint8 will have the value of 0x06 for PUSH (assigned in WAP WSP).
- Type-Specific Contents: provisioning document that must include HeadersLen, Content Type, Headers, Data, as defined below.
    - HeadersLen – value of 0x5A will be used for our first example which contains no headers, just Media-Type parameter. If the HeadersLen does contain the headers, its length will be accounted for here as well.
    - ContentType – media type parameter
    - Headers – optional parameter
    - Data – actual provisioning document

# Media Type Parameter

According to the WAP Provisioning Content specification, security information is "transported as parameters to the media type *application/vnd.wap.connectivity-wbxml*." The overall media-type parameter is formatted as follows:

- Mime-type; SEC=x; MAC=…
    - Mime-type: this will be application/vnd.wap.connectivity-wbxml
    - SEC: security mechanism. In this example, use the value 0x31 '1' – USERPIN
    - MAC: this is calculated using the WBXML document as the data and the user pin as the key for the HMAC calculation based on the SHA-1 algorithm. From the WAP Provisioning Bootstrap specification, "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 . . . This calculation is repeated in the ME when checking the validity of the MAC."

# String in Text Format

The following yields the string in text format (to be used in example):

```
application/vnd.wap.connectivity-wbxml;MAC=...;SEC=1
```

For this example, the User pin is "12345678" using the XML document from the "Provisioning Document" section below.

Hexadecimal dump:

```
# application/vnd.wap.connectivity-wbxml
61 70 70 6C  69 63 61 74  69 6F 6E 2F  76 6E 64 2E
77 61 70 2E  63 6F 6E 6E  65 63 74 69  76 69 74 79
2D 77 62 78  6D 6C

# ;
3B

# MAC=
4D 41 43 3D

# SHA1 HASH: 93B3 77A7 85A2 60C9 289E 36E0 F315 E051 9B2E D63F
39 33 42 33  37 37 41 37  38 35 41 32  36 30 43 39
32 38 39 45  33 36 45 30  46 33 31 35  45 30 35 31
39 42 32 45  44 36 33 46

# ;
3B
```

```
# SEC=1
53 45 43 3D  31

# NULL (End of string)
00
```

# String in Token Format

As an alternative, one could use token-format for the media-type parameter to save bytes. Since the Mime-type, SEC, and MAC are well known, they can be encoded as a short integer using the corresponding known assigned numbers.

The assigned number for application/vnd.wap.connectivity-wbxml is 0x36. This number will become b6 after short-integer encoding.

The assigned number for SEC is 0x11, it will become 0x91 after short-integer encoding. See reference [Wireless Specification Protocol] table 38. In this case, USERPIN (1) will encoded as 0x81.

The assigned number for MAC is 0x12, it will become 0x92 after short-integer encoding. See reference [Wireless Specification Protocol] table 38.

The MAC value (40 bytes) remains the same.

Hexadecimal dump of this will be:

```
# application/vnd.wap.connectivity-wbxml
b6

# SEC = 1
91 81

# MAC
92

# SHA1 HASH: 93B3 77A7 85A2 60C9 289E 36E0 F315 E051 9B2E D63F
same as before

# NULL (End of string for the encoded MAC value)
00
```

This token-format string will be demonstrated later in example #2.

## Headers (Optional Fields)

The operator may add headers following content type as part of Media-type parameters. In example #2, only one field, called “from” field (assigned number0x15, and therefore short-integer encoded as 0x95) which is used to specify sender’s email address is included. If the sender’s email is abc@mot.com, the Hexadecimal dump will be the following:

```
# From
95

#  abc@mot.com
61 62 63 40 6D 6F 74 2E 63 6F 6D

# NULL (End of string)
00
```

## Provisioning Document

In this provisioning example, the following XML document is used:

```
<?xml version="1.0"?>
<!DOCTYPE wap-provisioningdoc PUBLIC "-//WAPFORUM//DTD PROV 1.0//EN"
"http://www.wapforum.org/DTD/prov.dtd">
<wap-provisioningdoc>
<characteristic type="BOOTSTRAP">
      <parm name="PROXY-ID" value="sdp.our.com"/>
      <parm name="NAME" value="xyz"/>
</characteristic>
<characteristic type="PXLOGICAL">
      <parm name="PROXY-ID" value="sdp.our.com"/>
      <parm name="NAME" value="HelloWorld"/>
      <parm name="STARTPAGE"
value="http://sdp.our.com/IUT/homepage.htm"/>
      <characteristic type="PXPHYSICAL">
            <parm name="PHYSICAL-PROXY-ID" value="Gateway 1"/>
            <parm name="PXADDR" value="111.31.222.46"/>
            <parm name="PXADDRTYPE" value="IPV4"/>
            <parm name="TO-NAPID" value="NAP 1"/>
            <characteristic type="PORT">
                  <parm name="PORTNBR" value="80"/>
                  <parm name="SERVICE" value="OTA-HTTP-PO"/>
            </characteristic>
      </characteristic>
</characteristic>
<characteristic type="NAPDEF">
      <parm name="NAPID" value="NAP 1"/>
      <parm name="BEARER" value="GSM-GPRS"/>
      <parm name="NAME" value="OUR GPRS"/>
      <parm name="NAP-ADDRESS" value="our.net"/>
      <parm name="NAP-ADDRTYPE" value="APN"/>
</characteristic>
```

```
</wap-provisioningdoc>
```

The following table shows the translation into WBXML:

| Token String | Description |
|---|---|
| 03 | WBXML version 1.3 ,see [WBXML1] chapter 5.4 |
| 0B | The public identifier for "-//WAPFORUM//DTD PROV 1.0//EN" |
| 6A | Character set UTF-8. MIBEnum value is 106 which 6A in hex |
| 12 | String table length. |
| 73 64 70 2e 6f 75 72 2e 63 6f 6d 00 4e 41 50 20 31 00 | Offset 0: "sdp.our.com" and Offset 0C: "NAP 1" |
| 45 | wap-provisioningdoc with no version |
| C6 56 01 | Characteristic (06) BOOTSTRAP (56) with attribute and content ended with an end of the attribute list. |
| 87 15 06 | Parameter (07) PROXY-ID (15) with attribute but no content. The attribute is the value of 06. |
| 83 00 01 | The value is from the string table (83) from offset (00) "sdp.our.com" and end of the attribute list (01). |
| 87 07 06 | Parameter (07) NAME (07) with attribute but no content. The attribute is the value of 06 |
| 03 78 79 7a 00 01 | The value is inline string (03) "xyz" ending with the end of the string (00) and end of the attribute list (01). |
| 01 | End of the characteristic BOOTSTRAP. |
| C6 51 01 | Characteristic (06) PXLOGICAL (51) with attribute and content ended with an end of the attribute list. |
| 87 15 06 | Parameter (07) PROXY-ID (15) with attribute but no content. The attribute is the value of 06. |
| 83 00 01 | The value is from the string table (83) from offset (00) "sdp.our.com" and end of the attribute list (01). |
| 87 07 06 | Parameter (07) NAME (07) with attribute but no content. The attribute is the value of 06. |
| 03 48 65 6C 6C 6F 57 6F 72 6C 64 00 01 | The value is inline string (03) "HelloWorld" ending with the end of the string (00) and end of the attribute list (01). |
| 87 1C 06 | Parameter (07) STARTPAGE (1C) with attribute but no content. The attribute is the value of 06. |
| 03 68 74 74 70 3a 2f 2f 00 83 00 03 2f 49 55 54 2f 68 6f 6d 65 70 61 67 65 2e 68 74 6d 00 01 | The value is inline string (03) "http://" + string table (83) from offset (00) "sdp.our.com" + inline string (03) "/IUT/homepage.htm" ending with the end of the string (00) and end of the attribute list (01). |
| C6 52 01 | Characteristic (06) PXPHYSICAL (52) with attribute and content ended with an end of the attribute list. |
| 87 2F 06 | Parameter (07) PHYSICAL-PROXY-ID (2F) with attribute but no content. The |

| | |
|---|---|
| | attribute is the value of 06. |
| 03 47 61 74 65 77 61 79 20 31 00 01 | The value is inline string (03) “Gateway 1” ending with the end of the string (00) and end of the attribute list (01). |
| 87 20 06 | Parameter (07) PXADDR (20) with attribute but no content. The attribute is the value of 06. |
| 03 31 31 31 2e 33 31 2e 32 32 32 2e 34 36 00 01 | The value is inline string (03) “111.31.222.46” ending with the end of the string (00) and end of the attribute list (01). |
| 87 21 06 85 01 | Parameter (07) PXADDRTYPE (21) with attribute but no content. The attribute is the value of 06. The value of the attribute is “IP4” (85) with end of the attribute list. |
| 87 22 06 | Parameter (07) TO-NAPID (22) with attribute but no content. The attribute is the value of 06. |
| 83 0c 01 | The value is from the string table (83) from offset (0C) “NAP 1” and end of the attribute list (01). |
| C6 53 01 | Characteristic (06) PORT (53) with attribute and content ended with an end of the attribute list. |
| 87 23 06 | Parameter (07) PORTNBR (23) with attribute but no content. The attribute is the value of 06. |
| 03 38 30 00 01 | The value is inline string (03) “80” ending with the end of the string (00) and end of the attribute list (01). |
| 87 24 06 d2 01 | Parameter (07) SERVICE (24) with attribute but no content. The attribute is the value of 06. The value of attribute is OTA-HTTP-PO (d2) and end of attribute list (01). |
| 01 | End of the characteristic PORT. |
| 01 | End of the characteristic PXPHYSICAL. |
| 01 | End of the characteristic PXLOGICAL |
| c6 55 01 | Characteristic (06) NAPDEF (55) with attribute and content ended with an end of the attribute list. |
| 87 11 06 | Parameter (07) NAPID (11) with attribute but no content. The attribute is the value of 06. |
| 83 0c 01 | The value is from the string table (83) from offset (0C) “NAP 1” and end of the attribute list (01). |
| 87 10 06 AB 01 | Parameter (07) BEARER (10) with attribute but no content. The attribute is the value of 06. The value of attribute is GSM-GPRS (AB). |
| 87 07 06 | Parameter (07) NAME (07) with attribute but no content. The attribute is the value of 06. |
| 03 4F 55 52 20 47 50 52 53 00 01 | The value is inline string (03) “OUR GPRS” ending with the end of the string (00) and end of the attribute list (01). |
| 87 08 06 | Parameter (07) NAP-ADDRESS (08) with attribute but no content. The attribute is the value of 06. |
| 03 6F 75 72 2E 6E 65 74 00 01 | The value is inline string (03) “our.net” ending with the end of the string (00) and end of the attribute list (01). |

| | |
|---|---|
| 87 09 06 89 01 | Parameter (07) NAP-ADDRTYPE (09) with attribute but no content. The attribute is the value of 06. The value of the attribute is “APN” (89) with the end of the attribute list. |
| 01 | End of the characteristic NAPDEF. |
| 01 | End of the wap-provisioningdoc. |

The following is the hexadecimal dump of the WBXML file:

```
03 0b 6a 12 73 64 70 2e 6f 75 72 2e 63 6f 6d 00
4e 41 50 20 31 00 45 c6 56 01 87 15 06 83 00 01
87 07 06 03 78 79 7a 00 01 01 c6 51 01 87 15 06
83 00 01 87 07 06 03 48 65 6c 6c 6f 57 6f 72 6c
64 00 01 87 1c 06 03 68 74 74 70 3a 2f 2f 00 83
00 03 2f 49 55 54 2f 68 6f 6d 65 70 61 67 65 2e
68 74 6d 00 01 c6 52 01 87 2f 06 03 47 61 74 65
77 61 79 20 31 00 01 87 20 06 03 31 31 31 2e 33
31 2e 32 32 32 2e 34 36 00 01 87 21 06 85 01 87
22 06 83 0c 01 c6 53 01 87 23 06 03 38 30 00 01
87 24 06 d2 01 01 01 01 c6 55 01 87 11 06 83 0c
01 87 10 06 ab 01 87 07 06 03 4f 55 52 20 47 50
52 53 00 01 87 08 06 03 6f 75 72 2e 6e 65 74 00
01 87 09 06 89 01 01 01
```

## Adapting to GSM SMS Format

The following is an explanation of the header format for each message:

1. SMSC Number – HelloWorld SMSC

Example: 112233008000

- Length: (number of bytes including TON/NPI) 07
- TON/NPI: 91
- MSISDN: 11 22 33 00 08 00 FF FF FF FF FF

2. TPDU Length (variable – one byte, the sum of all remaining items)

Value = xx (varies)

3. SMS Deliver TPDU (one byte)

Value = 40

- TP-MTI: (bits 1 and 0) ‘00’ SMS-DELIVER
- TP-MMS: (bit 2) ‘0’ more messages are waiting (1st and 2nd message)
- TP-RP: (bit 7) ‘0’ reply path NOT present
- TP-UDHI: (bit 6) ‘1’ User Data contains a header

- TP-SRI: (bit 5) ‘0’ Status report will not be returned
- Not used: (bits 4 and 3) ‘00’

4. TP-OA (variable – the originating address)

Example: Number 0123987654

- Length: (number of digits excluding TON/NPI) 0A
- TON/NPI: 81
- MSISDN: 10 32 89 67 45

6. TP-PID

Value: 00

- Usage: (bits 7 and 6) ‘00’ Assigns bits 0…5 as defined below
- Telematic networking: (bit 5) ‘0’ no interworking
- Telematic device: (bits 4-0) ‘00000’ not used since bit 5 is ‘0’

7. TP-DCS

Value: F5

- Coding group bits: (bits 7-4) ‘1111’
- Reserved: (bit 3) ‘0’
- Message coding: (bit 2) ‘1’ 8-bit data
- Message class: (bits 1 and 0) ‘01’ class 1, default meaning

8. TP-SCTS (7 bytes)

Value: 10 20 72 61 54 85 00 (example)

9. TP-UDL

Value: 8C (maximum 140 bytes)

10. TP-UD

A. TP-UDH – 12-byte header for each message

Example: 0B 05 04 0B 84 00 00 00 03 00 03 01

- UDH Length: 0B
- PORT Addressing
    - IE Type: 05 16-bit port addressing
    - IE Length: 04
    - Destination Port: 0B 84 (2948 – push connectionless port)
    - Source Port : 00 00 (0000 – does not matter)

- SAR
    - IE Type: 00 SAR
    - IE Length: 03
    - Datagram Reference Number: 00
    - Total number of segments: 03 (three segments)
    - Current segment number: 01 (first segment)

B. TP-UD – Binary data (WBXML file) Maximum 128 bytes

Hexadecimal dump for each message:

The total length of our provisioning data is 309 bytes:

- TID – 1 byte
- PDU Type – 1 byte
- Header Length – 1 byte
- Content Type Length – 90 bytes
- Headers – 0 bytes
- Provisioning document – 216 bytes

Since each message can only carry 128 bytes of data, this will require at least three SMS messages. We will illustrate this in the next section.

# Examples

Example #1 – text format (in first segment)

- *Message 1*

```
07 91 11 22 33 00 80 00 FF FF FF FF FF // SMSC
9E // TPDU LENGTH
40 // SMS Deliver TPDU (more messages waiting)
0A 81 10 32 89 67 45 // TP-OA
00 // TP-PID
F5 // TP-DCS
10 20 72 61 54 85 00 // TP-SCTS
8C //TP-UDL
0B 05 04 0B 84 00 00 00 03 00 03 01 //TP-UDH
// TP-UD
01 // TID
06 // UNIT PUSH
5A // HEADER LENGTH
// MEDIA-TYPE PARAMETER
61 70 70 6C  69 63 61 74  69 6F 6E 2F  76 6E 64 2E
77 61 70 2E  63 6F 6E 6E  65 63 74 69  76 69 74 79
2D 77 62 78  6D 6C 3B 53  45 43 3D 31  3B 4D 41 43
3D 39 33 42  33 37 37 41  37 38 35 41  32 36 30 43
39 32 38 39  45 33 36 45  30 46 33 31  35 45 30 35
```

```
31 39 42 32  45 44 36 33  46 00
// WBXML (BYTES 1 – 35 / 216)
03 0b 6a 12  73 64 70 2e  6f 75 72 2e  63 6f 6d 00
4e 41 50 20  31 00 45 c6  56 01 87 15  06 83 00 01
87 07 06
```

- *Message 2*

```
07 91 11 22 33 00 80 00 FF FF FF FF FF // SMSC
9E // TPDU LENGTH
40 // SMS Deliver TPDU (more messages waiting)
0A 81 10 32 89 67 45 // TP-OA
00 // TP-PID
F5 // TP-DCS
10 20 72 61 54 85 00 // TP-SCTS
8C //TP-UDL
0B 05 04 0B 84 00 00 00 03 00 03 02 //TP-UDH
// TP-UD
// WBXML (BYTES 36 – 163 / 216)
03 78 79 7a  00 01 01 c6  51 01 87 15  06 83 00 01
87 07 06 03  48 65 6c 6c  6f 57 6f 72  6c 64 00 01
87 1c 06 03  68 74 74 70  3a 2f 2f 00  83 00 03 2f
49 55 54 2f  68 6f 6d 65  70 61 67 65  2e 68 74 6d
00 01 c6 52  01 87 2f 06  03 47 61 74  65 77 61 79
20 31 00 01  87 20 06 03  31 31 31 2e  33 31 2e 32
32 32 2e 34  36 00 01 87  21 06 85 01  87 22 06 83
0c 01 c6 53  01 87 23 06  03 38 30 00  01 87 24 06
```

- *Message 3*

```
07 91 11 22 33 00 80 00 FF FF FF FF FF // SMSC
53 // TPDU LENGTH
44 // SMS Deliver TPDU (no more messages waiting)
0A 81 10 32 89 67 45 // TP-OA
00 // TP-PID
F5 // TP-DCS
10 20 72 61 54 85 00 // TP-SCTS
41 //TP-UDL
0B 05 04 0B 84 00 00 00 03 00 03 03 //TP-UDH
// TP-UD
// WBXML (BYTES 164 – 216 / 216)
d2 01 01 01  01 c6 55 01  87 11 06 83  0c 01 87 10
06 ab 01 87  07 06 03 4f  55 52 20 47  50 52 53 00
01 87 08 06  03 6f 75 72  2e 6e 65 74  00 01 87 09
06 89 01 01  01
```

Example #2 – token format (in first segment), and including headers

- *Message 1*

```
07 91 11 22 33 00 80 00 FF FF FF FF FF // SMSC
80 // TPDU LENGTH
40 // SMS Deliver TPDU (more messages waiting)
0A 81 10 32 89 67 45 // TP-OA
00 // TP-PID
F5 // TP-DCS
10 20 72 61 54 85 00 // TP-SCTS
6E //TP-UDL
0B 05 04 0B 84 00 00 00 03 00 03 01 //TP-UDH
// TP-UD
```

```
01 // TID
06 // UNIT PUSH
3C // Length for content-type and HEADERS
1F 2D  // Content type value length given as “Length-quote Length”

// MEDIA-TYPE PARAMETER
B6  // for application/vnd.wap.connectivity-wbxml
91 81  // for SEC and USERPIN (1)
92 // MAC
39 33 42 33  37 37 41 37  38 35 41 32  36 30 43 39
32 38 39 45  33 36 45 30  46 33 31 35  45 30 35 31
39 42 32 45  44 36 33 46  00
95  // From field in headers
61 62 63 40  6D 6F 74 2E  63 6F 6D 00 // abc@mot.com

// WBXML (BYTES 1 – 35 / 216)
03 0b 6a 12  73 64 70 2e  6f 75 72 2e  63 6f 6d 00
4e 41 50 20  31 00 45 c6  56 01 87 15  06 83 00 01
87 07 06
```

NOTE: 43 bytes are saved compared with the method used in example #1. Message 2 and Message 3 are the same as that in example #1.

# Appendix A: Parameter Mapping

Browser:

| Prompt | Description | OMA Parameter |
|---|---|---|
| Name | Name of the Internet Setup | PXLOGICAL / NAME |
| Homepage | The homepage for this session | PXLOGICAL / STARTPAGE |
| Service Type 1 | The service offered by Port 1 | SERVICE[1] |
| Gateway IP 1 | Primary proxy address | PXPHYSICAL / PXADDR |
| Port 1 | Port number to be used for proxy 1 | PORT[1] |
| Domain 1 | The domain covered by proxy 1 | DOMAIN[1] |
| Service Type 2 | The service offered by Port 2 | SERVICE[1] |
| Gateway IP 2 | Secondary proxy address | PXPHYSICAL / PXADDR |
| Port 2 | Port number to be used for proxy 2 | PORT[1] |
| Domain 2 | The domain covered by proxy 2 | DOMAIN[1] |
| DNS 1 | Primary DNS IP Address | NAPDEF / DNS-ADDR |
| DNS 2 | Secondary DNS IP Address | NAPDEF / DNS-ADDR |
| CSD Timeout | CSD Linger Timer | NAPDEF / LINGER |
| CSD No. 1 | Primary phone number according to E164 Scheme | NAPDEF / NAP-ADDRESS |
| User Name 1 | User Name for CSD No. 1 login | NAPAUTHINFO / AUTHNAME |
| Password 1 | Password for CSD No. 1 login | NAPAUTHINFO / AUTHSECRET |
| Speed (Bps) 1 | Speed for CSD No. 1 | NAPDEF / LINKSPEED |
| Line Type 1 | Call Type for CSD No. 1 | NAPDEF / NAP-ADDRTYPE |
| CSD No. 2 | Secondary phone number according to E164 Scheme | NAPDEF / NAP-ADDRESS |
| User Name 2 | User Name for CSD No. 2 login | NAPAUTHINFO / AUTHNAME |
| Password 2 | Password for CSD No. 2 login | NAPAUTHINFO / AUTHSECRET |
| Speed (Bps) 2 | Speed for CSD No. 2 | NAPDEF / LINKSPEED |
| Line Type 2 | Call Type for CSD No. 2 | NAPDEF / NAP-ADDRTYPE |
| GPRS APN | APN for Packet Data Access | NAPDEF / NAP-ADDRESS |
| User Name | User name for CHAP/PAP GPRS APN login | NAPAUTHINFO / AUTHNAME |
| Password | Password for CHAP/PAP GPRS APN login | NAPAUTHINFO / AUTHSECRET |

[1]Present in either PXLOGICAL or PXPHYSICAL – PXPHYSICAL has higher priority.

MMS:

| Prompt | Description | OMA Parameter |
|---|---|---|
| Service Name | MMS session name, same as the related browser session | APPLICATION/NAME |
| Server Name | MMS server address | APPLICATION/ADDR |
| Web Session Name | The corresponding browser session name | PXLOGICAL/NAME |

SyncML Data Synchronization:

| Prompt | Description | OMA Parameter |
|---|---|---|
| Name | SyncML DS session name | APPLICATION/NAME |
| URL | SyncML DS Server address | APPLICATION/ADDR |
| User Name | User name for SyncML DS session login | APPAUTH/AAUTHNAME |
| Password | Password for SyncML DS session login | APPAUTH/AAUTHSECRET |
| Phonebook | Contacts data resource path | RESOURCE/URI |
| Datebook | Appointments data resource path | RESOURCE/URI |

Email:

| Prompt | Description | OMA Parameter |
|---|---|---|
| Account Name | Name of Email Account to be configured | APPLICATION / NAME |
| User ID (Receiving User ID[1]) | User ID used for authentication on server | APPAUTH / AAUTHNAME |
| Password (Receiving | Password used for authentication on server | APPAUTH / AAUTHSECRET |
| Sending User ID[1] | User ID used for authentication on sending server | APPAUTH / AAUTHNAME |
| Sending Password[1] | Password used for authentication on sending server | APPAUTH / AAUTHSECRET |
| Return Address | Return address used for "From:" field in email messages | APPLICATION/FROM |
| Protocol | Shows receiving protocol (IMAP4, POP3) | APPLICATION/APPID[2] |
| Sending host | Domain name or IP address for sending server | APPLICATION/ADDR or APPADDR/ADDR |
| Sending port | Port name used to connect for SMTP protocol | 25 or PORT/PORTNBR |
| Receiving host | Domain name or IP address for receiving server | APPLICATION/ADDR or APPADDR/ADDR |
| Receiving port | Port name used to connect for POP3/IMAP4 protocol | 110/143 or PORT/PORTNBR |
| Security->Use SSL for sending | Use SSL for sending operations | PORT/SERVICE |
| Security->Use SSL for receiving | Use SSL for receiving operations | PORT/SERVICE |
| Web Session | Web Session name used to establish connection | NAPDEF/NAME pointed by APPLICATION/TO-NAPID |
| | | |
| Web Session Name | The corresponding browser session name | NAPDEF / NAME |
| DNS 1 | Primary DNS IP Address | NAPDEF / DNS-ADDR |
| DNS 2 | Secondary DNS IP Address | NAPDEF / DNS-ADDR |
| CSD No. 1 | Primary phone number according to E164 Scheme | NAPDEF / NAP-ADDRESS |
| User Name 1 | User Name for CSD No. 1 login | NAPAUTHINFO / AUTHNAME |
| Password 1 | Password for CSD No. 1 login | NAPAUTHINFO / AUTHSECRET |
| GPRS APN | APN for Packet Data Access | NAPDEF / NAP-ADDRESS |
| User Name | User name for CHAP/PAP GPRS APN login | NAPAUTHINFO / AUTHNAME |
| Password | Password for CHAP/PAP GPRS APN login | NAPAUTHINFO / AUTHSECRET |

[1] This parameter used only if “Separate Credentials for SMTP” is enabled for Email
[2] If parameter equals to 110 or 143

# Appendix B: Compliancy Matrix

Please note the following compliancy matrix is for products using the Motorola Internet Browser (MIB) version 2.2 and later. A partial listing of the browser version for popular phone models is provided in Appendix C.

| Character Set and Encoding | | | | |
|---|---|---|---|---|
| Item # | Function | Ref | Status | Motorola |
| ProvCont-CSE-C-001 | UTF-8 Encoding | 4.7 | M | Y |
| ProvCont-CSE-C-002 | Character entities | 4.7 | M | Y |

| Content Format and Tokenization | | | | |
|---|---|---|---|---|
| Item # | Function | Ref | Status | Motorola |
| ProvCont-CO-C-001 | Support for the WAP-Provisioning-doc DTD | 4.1 | M | Y |
| ProvCont-CO-C-002 | Support for the WAP-Provisioning-doc DTD or textual form (text/vnd.wap.connectivity-xml) | 4.2 | O | N |
| ProvCont-CO-C-003 | Support for the WAP-Provisioning-doc DTD in tokenized form (application/vnd.wap.connectivity.wbxml) | 7 | M | Y<br>Up to 10 provisioning profiles are supported. |
| ProvCont-CO-C-004 | Support for media type parameter MAC | 4.3 | M | Y |
| ProvCont-CO-C-005 | Support for media type parameter SEC | 4.3 | M | Y |

| Elements and Attributes | | | | |
|---|---|---|---|---|
| Item # | Function | Ref | Status | Motorola |
| ProvCont-CEA-C-001 | Support for the element wap-provisioningdoc | 4.4 | M | Y |
| ProvCont-CEA-C-002 | Support for the element characteristic | 4.5 | M | Y |
| ProvCont-CEA-C-003 | Support for the element parm | 4.6 | M | Y |
| ProvCont-CEA-C-004 | Support for the wap-provisioningdoc attribute “version” | 4.4 | M | Y |
| ProvCont-CEA-C-005 | Support for the characteristic attribute “type” | 4.5 | M | Y |
| ProvCont-CEA-C-006 | Support for the parm attribute “name” | 4.6 | M | Y |
| ProvCont-CEA-C-007 | Support for the parm attribute “value” | 4.6 | M | Y |

| Characteristics | | | | |
|---|---|---|---|---|
| Item # | Function | Ref | Status | Motorola |
| ProvCont-CC-C-001 | Support for the characteristic PXLOGICAL | 4.5.1 | M | Y<br>Up to 10 PXLOGICALs are supported. |
| ProvCont-CC-C-002 | Support for the characteristic PXPHYSICAL | 4.5.2 | M | Y<br>Two characteristics per PXLOGICAL will be supported. |
| ProvCont-CC-C-003 | Support for the characteristic PXAUTHINFO | 4.5.3 | O | N |
| ProvCont-CC-C-004 | Support for the characteristic NAPDEF | 4.5.5 | M | Y<br>Two CSD and one GPRS NAPDEF per provisioning document will be supported. |

| | | | | |
|---|---|---|---|---|
| | | | | For Email: only one NAPDEF per account is supported. |
| ProvCont-CC-C-005 | Support for the characteristic NAPAUTHINFO | 4.5.6 | O | Y |
| ProvCont-CC-C-006 | Support for the characteristic PORT | 5.2<br>4.5.4 | M | Y |
| ProvCont-CC-C-007 | Support for the characteristic VALIDITY | 4.5.7 | O | N |
| ProvCont-CC-C-008 | Support for the characteristic BOOTSTRAP | 4.5.8 | O | Y<br>We don't support continuous provisioning |
| ProvCont-CC-C-00 | Support for the characteristic CLIENTIDENTITY | 4.5.9 | O | N |
| ProvCont-CC-C-010 | Support for the characteristic VENDORCONFIG | 4.5.10 | O | N |
| ProvCont-CC-C-011 | Support for the characteristic APPLICATION | 4.5.11 | O | Y<br>Limited use for MMS, SyncML Data Synchronization & Email |
| ProvCont-CC-C-012 | Support for the characteristic APPADDR | 4.5.12 | O | Y<br>Limited use for Email |
| ProvCont-CC-C-013 | Support for the characteristic APPAUTH | 4.5.13 | O | Y<br>Limited use for SyncML Data Synchronization & Email |
| ProvCont-CC-C-014 | Support for the characteristic RESOURCE | 4.5.14 | O | Y<br>Limited use for SyncML Data Synchronization |
| ProvCont-CC-C-015 | Support for the characteristic ACCESS | 4.5.15 | O | N |

| Characteristic PXLOGICAL | | | | |
|---|---|---|---|---|
| Item # | Function | Ref | Status | Motorola |

| | | | | |
|---|---|---|---|---|
| ProvCont-CPL-C-001 | Support for the parm PROXY-ID | 4.6.1 | M | Y |
| ProvCont-CPL-C-002 | Support for the parm PROXY-PROVIDER-ID | 4.6.1 | O | N |
| ProvCont-CPL-C-003 | Support for the parm NAME | 4.6.1 | M | Y |
| ProvCont-CPL-C-004 | Support for the parm DOMAIN | 4.6.1 | M | Y |
| ProvCont-CPL-C-005 | Support for the parm TRUST | 4.6.1 | O | N |
| ProvCont-CPL-C-006 | Support for the parm MASTER | 4.6.1 | O | N |
| ProvCont-CPL-C-007 | Support for the parm STARTPAGE | 4.6.1 | M | Y |
| ProvCont-CPL-C-008 | Support for the parm BASAUTH-ID | 4.6.1 | M | N<br>The preferred approach is to have this entered by the user to do proxy authentication based on a challenge from the proxy. |
| ProvCont-CPL-C-009 | Support for the parm BASAUTH-PW | 4.6.1 | M | |
| ProvCont-CPL-C-010 | Support for the parm WSP-VERSION | 4.6.1 | O | N |
| ProvCont-CPL-C-011 | Support for the parm PUSHENABLED | 4.6.1 | O | N |
| ProvCont-CPL-C-012 | Support for PORT characteristic within PXLOGICAL | 4.5.4 | M | Y |
| ProvCont-CPL-C-013 | Support for multiple PORT characteristics within PXLOGICAL | 4.5.4 | O | N |
| ProvCont-CPL-C-014 | Support for parm PROXY-PW | 4.6.1 | O | N |
| ProvCont-CPL-C-015 | Support for parm PPGAUTH-TYPE | 4.6.1 | O | N |

| | | | | |
|---|---|---|---|---|
| ProvCont-CPL-C-016 | Support for parm PULLENABLED | 4.6.1 | O | N |

| Characteristic PXPHYSICAL | | | | |
|---|---|---|---|---|
| Item # | Function | Ref | Status | Motorola |
| ProvCont-CPP-C-001 | Support for the parm PHYSICAL-PROXY-ID | 4.6.2 | M | Y |
| ProvCont-CPP-C-002 | Support for the parm PXADDR | 4.6.2 | M | Y (Motorola Browser supports only IPV4) |
| ProvCont-CPP-C-003 | Support for the parm PXADDRTYPE | 4.6.2 | M | Y |
| ProvCont-CPP-C-004 | Support for the parm TO-NAPID | 4.6.2 | M | Y |
| ProvCont-CPP-C-005 | Support for the parm DOMAIN | 4.6.2 | O | Y |
| ProvCont-CPP-C-006 | Support for the parm WSP-VERSION | 4.6.2 | O | N |
| ProvCont-CPP-C-007 | Support for the parm PUSHENABLED | 4.6.2 | O | N |
| ProvCont-CPP-C-008 | Support for the TO-NAPID value “INTERNET” | 4.6.2 | O | N |
| ProvCont-CPP-C-009 | Support for the PXADDRTYPE value “IPV4” | 4.6.2 | O | Y |
| ProvCont-CPP-C-010 | Support for the PXADDRTYPE value “IPV6” | 4.6.2 | O | N |
| ProvCont-CPP-C-011 | Support for the PXADDRTYPE value “E164” | 4.6.2 | O | N |
| ProvCont-CPP-C-012 | Support for the PXADDRTYPE value “ALPHA” | 4.6.2 | O | N |
| ProvCont-CPP-C-013 | Support for PORT characteristics within PXPHYSICAL | 4.5.4 | M | Y |

| | | | | |
|---|---|---|---|---|
| ProvCont-CPP-C-014 | Support for multiple PORT characteristics within PXPHYSICAL | 4.5.4 | O | N |
| ProvCont-CPP-C-015 | Support for multiple TO-NAPID within one PXPHYSICAL | 4.6.2 | O | Y<br>However, the NAPIDs are applied to all characteristics within the PXLOGICAL document. Not per PXPHYSICAL characteristic. |
| ProvCont-CPP-C-016 | Support for the parm PXADDR-FQDN | 4.6.2 | O | N |
| ProvCont-CPP-C-017 | Support for the parm PULLENABLED | 4.6.2 | O | N |

| Characteristic PXAUTHINFO | | | | |
|---|---|---|---|---|
| Item # | Function | Ref | Status | Motorola |
| ProvCont-CPA-C-001 | Support for the parm PXAUTH-TYPE | 4.6.3 | O | N |
| ProvCont-CPA-C-002 | Support for the parm PXAUTH-ID | 4.6.3 | O | N |
| ProvCont-CPA-C-003 | Support for the parm PXAUTH-PW | 4.6.3 | O | N |
| ProvCont-CPA-C-004 | Support for PXAUTH-TYPE value “HTTP-BASIC” | 4.6.3 | O | N |
| ProvCont-CPA-C-005 | Support for PXAUTH-TYPE value “HTTP-DIGEST” | 4.6.3 | O | N |
| ProvCont-CPA-C-006 | Support for PXAUTH-TYPE value “WTLS-SS” | 4.6.3 | O | N |

| Characteristic PORT | | | | |
|---|---|---|---|---|
| Item # | Function | Ref | Status | Motorola |
| ProvCont-CP-C-001 | Support for the parm PORTNBR | 4.6.4 | M | Y |
| ProvCont-CP-C-002 | Support for the parm SERVICE | 4.6.4 | M | Y |

| | | | | |
|---|---|---|---|---|
| ProvCont-CP-C-003 | Support for SERVICE value “CL-WSP” | 4.6.4 | O | Y<br>(These parameters are defined only for WAP Pull) |
| ProvCont-CP-C-004 | Support for SERVICE value “CO-WSP” | 4.6.4 | O | Y<br>(These parameters are defined only for WAP Pull) |
| ProvCont-CP-C-005 | Support for SERVICE value “CL-SEC-WSP” | 4.6.4 | O | Y<br>(These parameters are defined only for WAP Pull) |
| ProvCont-CP-C-006 | Support for SERVICE value “CO-SEC-WSP” | 4.6.4 | O | Y<br>(These parameters are defined only for WAP Pull) |
| ProvCont-CP-C-007 | Support for SERVICE value “CO-SEC-WTA” | 4.6.4 | O | N |
| ProvCont-CP-C-008 | Support for SERVICE value “CL-SEC-WTA” | 4.6.4 | O | N |
| ProvCont-CP-C-009 | Support for SERVICE value “OTA-HTTP-TO” | 4.6.4 | O | N |
| ProvCont-CP-C-010 | Support for SERVICE value “OTA-HTTP-TLS-TO” | 4.6.4 | O | N |
| ProvCont-CP-C-011 | Support for SERVICE value “OTA-HTTP-PO” | 4.6.4 | O | Y |
| ProvCont-CP-C-012 | Support for SERVICE value “OTA-HTTP-TLS-PO” | 4.6.4 | O | Y |

| Characteristic NAPDEF | | | | |
|---|---|---|---|---|
| Item # | Function | Ref | Status | Motorola |
| ProvCont-CND-C-001 | Support for the parm NAPID | 4.6.5 | M | Y |
| ProvCont-CND-C-002 | Support for the parm BEARER | 4.6.5 | M | Y |
| ProvCont-CND-C-003 | Support for the parm NAME | 4.6.5 | M | Y |

| | | | | |
|---|---|---|---|---|
| ProvCont-CND-C-004 | Support for the parm INTERNET | 4.6.5 | O | N |
| ProvCont-CND-C-005 | Support for the parm NAP-ADDRESS | 4.6.5 | M | Y |
| ProvCont-CND-C-006 | Support for the parm NAP-ADDRTYPE | 4.6.5 | M | Y |
| ProvCont-CND-C-007 | Support for the parm CALLTYPE | 4.6.5 | O | Y |
| ProvCont-CND-C-008 | Support for the parm LOCAL-ADDR | 4.6.5 | O | N |
| ProvCont-CND-C-009 | Support for the parm LOCAL-ADDRTYPE | 4.6.5 | O | N |
| ProvCont-CND-C-010 | Support for the parm LINKSPEED | 4.6.5 | O | Y |
| ProvCont-CND-C-011 | Support for the parm DNLINKSPEED | 4.6.5 | O | N |
| ProvCont-CND-C-012 | Support for the parm LINGER | 4.6.5 | O | Y |
| ProvCont-CND-C-013 | Support for the parm DELIVERY-ERR-SDU | 4.6.5 | O | N |
| ProvCont-CND-C-014 | Support for the parm DELIVERY-ORDER | 4.6.5 | O | N |
| ProvCont-CND-C-015 | Support for the parm TRAFFIC-CLASS | 4.6.5 | O | N |
| ProvCont-CND-C-016 | Support for the parm MAX-SDU-SIZE | 4.6.5 | O | N |
| ProvCont-CND-C-017 | Support for the parm MAX-BITRATE-UPLINK | 4.6.5 | O | N |
| ProvCont-CND-C-018 | Support for the parm MAX-BITRATE-DNLINK | 4.6.5 | O | N |
| ProvCont-CND-C-019 | Support for the parm RESIDUAL-BER | 4.6.5 | O | N |

| | | | | |
|---|---|---|---|---|
| ProvCont-CND-C-020 | Support for the parm SDU-ERROR-RATIO | 4.6.5 | O | N |
| ProvCont-CND-C-021 | Support for the parm TRAFFIC-HANDL-PRIO | 4.6.5 | O | N |
| ProvCont-CND-C-022 | Support for the parm TRANSFER-DELAY | 4.6.5 | O | N |
| ProvCont-CND-C-023 | Support for the parm GUARANTEED-BITRATE-UPLINK | 4.6.5 | O | N |
| ProvCont-CND-C-024 | Support for the parm GUARANTEED-BITRATE-DNLINK | 4.6.5 | O | N |
| ProvCont-CND-C-025 | Support for multiple BEARER within one NAPDEF | 4.6.5 | O | N |
| ProvCont-CND-C-026 | Support for NAP-ADDRTYPE value “IPV4” | 4.6.5 | O | N |
| ProvCont-CND-C-027 | Support for NAP-ADDRTYPE value “IPV6” | 4.6.5 | O | N |
| ProvCont-CND-C-028 | Support for NAP-ADDRTYPE value “E164” | 4.6.5 | O | Y |
| ProvCont-CND-C-029 | Support for NAP-ADDRTYPE value “ALPHA” | 4.6.5 | O | N |
| ProvCont-CND-C-030 | Support for NAP-ADDRTYPE value “APN” | 4.6.5 | O | Y |
| ProvCont-CND-C-031 | Support for NAP-ADDRTYPE value “SCODE” | 4.6.5 | O | N |
| ProvCont-CND-C-032 | Support for NAP-ADDRTYPE value “TETRA-ITSI” | 4.6.5 | O | N |
| ProvCont-CND-C-033 | Support for NAP-ADDRTYPE value “MAN” | 4.6.5 | O | N |
| ProvCont-CND-C-034 | Support for CALLTYPE value “ANALOG-MODEM” | 4.6.5 | O | Y |

| | | | | |
|---|---|---|---|---|
| ProvCont-CND-C-035 | Support for CALLTYPE value “V.120” | 4.6.5 | O | N |
| ProvCont-CND-C-036 | Support for CALLTYPE value “V.110” | 4.6.5 | O | N |
| ProvCont-CND-C-037 | Support for CALLTYPE value “X.31” | 4.6.5 | O | N |
| ProvCont-CND-C-038 | Support for CALLTYPE value “BIT-TRANSPARENT” | 4.6.5 | O | N |
| ProvCont-CND-C-039 | Support for CALLTYPE value “DIRECT-ASYNCHRONOUS-DATA-SERVICE” | 4.6.5 | O | Y |
| ProvCont-CND-C-040 | Support for LOCAL-ADDRTYPE value “IPV4” | 4.6.5 | O | N |
| ProvCont-CND-C-041 | Support for LOCAL-ADDRTYPE value “IPV6” | 4.6.5 | O | N |
| ProvCont-CND-C-042 | Support for the parm DNS-ADDR | 4.6.5 | O | Y |
| ProvCont-CND-C-043 | Support for the parm MAX-NUMRETRY | 4.6.5 | O | N |
| ProvCont-CND-C-044 | Support for the parm FIRST-RETRYTIMEOUT | 4.6.5 | O | N |
| ProvCont-CND-C-045 | Support for the parm REREGTHRESHOLD | 4.6.5 | O | N |
| ProvCont-CND-C-046 | Support for the parm T-BIT | 4.6.5 | O | N |

| Bearers supported within NAPDEF Characteristics | | | | |
|---|---|---|---|---|
| Item # | Function | Ref | Status | Motorola |
| ProvCont-CBS-C-001 | Support for the BEARER value “GSM-USSD” | 4.6.5 | O | N |
| ProvCont-CBS-C-002 | Support for the BEARER value “GSM-SMS” | 4.6.5 | O | N<br>(These parameters are defined only for WAP Pull |

| | | | | |
|---|---|---|---|---|
| | | | | over SMS) |
| ProvCont-CBS-C-003 | Support for the BEARER value “ANSI-136-GUTS” | 4.6.5 | O | Y |
| ProvCont-CBS-C-004 | Support for the BEARER value “IS-95-CDMA-SMS” | 4.6.5 | O | N<br>(These parameters are defined only for WAP Pull over SMS) |
| ProvCont-CBS-C-005 | Support for the BEARER value “IS-95-CDMA-CSD” | 4.6.5 | O | Y |
| ProvCont-CBS-C-006 | Support for the BEARER value “IS-95-CDMA-PACKET” | 4.6.5 | O | Y |
| ProvCont-CBS-C-007 | Support for the BEARER value “ANSI-136-CSD” | 4.6.5 | O | Y |
| ProvCont-CBS-C-008 | Support for the BEARER value “ANSI-136-GPRS” | 4.6.5 | O | Y |
| ProvCont-CBS-C-009 | Support for the BEARER value “GSM-CSD” | 4.6.5 | O | Y |
| ProvCont-CBS-C-010 | Support for the BEARER value “GSM-GPRS” | 4.6.5 | O | Y |
| ProvCont-CBS-C-011 | Support for the BEARER value “AMPS-CDPD” | 4.6.5 | O | N |
| ProvCont-CBS-C-012 | Support for the BEARER value “PDC-CSD” | 4.6.5 | O | N |
| ProvCont-CBS-C-013 | Support for the BEARER value “PDC-PACKET” | 4.6.5 | O | N |
| ProvCont-CBS-C-014 | Support for the BEARER value “IDEN-SMS” | 4.6.5 | O | N |
| ProvCont-CBS-C-015 | Support for the BEARER value “IDEN-CSD” | 4.6.5 | O | N |
| ProvCont-CBS-C-016 | Support for the BEARER value “IDEN-PACKET” | 4.6.5 | O | N |

| | | | | |
|---|---|---|---|---|
| ProvCont-CBS-C-017 | Support for the BEARER value “FLEX/REFLEX” | 4.6.5 | O | N |
| ProvCont-CBS-C-018 | Support for the BEARER value “PHS-SMS” | 4.6.5 | O | N |
| ProvCont-CBS-C-019 | Support for the BEARER value “PHS-CSD” | 4.6.5 | O | N |
| ProvCont-CBS-C-020 | Support for the BEARER value “TETRA-SDS” | 4.6.5 | O | N |
| ProvCont-CBS-C-021 | Support for the BEARER value “TETRA-PACKET” | 4.6.5 | O | N |
| ProvCont-CBS-C-022 | Support for the BEARER value “ANSI-136-GHOST” | 4.6.5 | O | N |
| ProvCont-CBS-C-023 | Support for the BEARER value “MOBITEX-MPAK” | 4.6.5 | O | N |
| ProvCont-CBS-C-024 | Support for the BEARER value “CDMA2000-1X-SIMPLE-IP” | 4.6.5 | O | N |
| ProvCont-CBS-C-025 | Support for the BEARER value “CDMA2000-1X-MOBILE-IP” | 4.6.5 | O | N |

| Characteristic NAPAUTHINFO | | | | |
|---|---|---|---|---|
| Item # | Function | Ref | Status | Motorola |
| ProvCont-CAN-C-001 | Support for the parm AUTHTYPE | 4.6.6 | O | Y |
| ProvCont-CAN-C-002 | Support for the parm AUTHNAME | 4.6.6 | O | Y |
| ProvCont-CAN-C-003 | Support for the parm AUTHSECRET | 4.6.6 | O | Y |
| ProvCont-CAN-C-004 | Support for AUTHTYPE value “PAP” | 4.6.6 | O | Y |
| ProvCont-CAN-C-005 | Support for AUTHTYPE value “CHAP” | 4.6.6 | O | Y |

| | | | | |
|---|---|---|---|---|
| ProvCont-CAN-C-006 | Support for AUTHTYPE value “MD5” | 4.6.6 | O | N |
| ProvCont-CAN-C-007 | Support for parm AUTH-ENTITY value | 4.6.6 | O | N |
| ProvCont-CAN-C-008 | Support for parm SPI | 4.6.6 | O | N |
| ProvCont-CAN-C-009 | Support for AUTH-ENTITY value “AAA” | 4.6.6 | O | N |
| ProvCont-CAN-C-010 | Support for AUTH-ENTITY value “HA” | 4.6.6 | O | N |

| Characteristic Validity | | | | |
|---|---|---|---|---|
| Item # | Function | Ref | Status | Motorola |
| ProvCont-CV-C-001 | Support for the parm COUNTRY | 4.6.7 | O | N |
| ProvCont-CV-C-002 | Support for the parm NETWORK | 4.6.7 | O | N |
| ProvCont-CV-C-003 | Support for the parm SID | 4.6.7 | O | N |
| ProvCont-CV-C-004 | Support for the parm SOC | 4.6.7 | O | N |
| ProvCont-CV-C-005 | Support for the parm VALIDUNTIL | 4.6.7 | O | N |
| ProvCont-CV-C-006 | Support for multiple MNC in NETWORK value field | 4.6.7 | O | N |
| ProvCont-CV-C-007 | Support for multiple SID in SID value field | 4.6.7 | O | N |

| Characteristic BOOTSTRAP | | | | |
|---|---|---|---|---|
| Item # | Function | Ref | Status | Motorola |
| ProvCont-CB-C-001 | Support for the parm PROVURL | 4.6.8 | O | N |

| | | | | |
|---|---|---|---|---|
| ProvCont-CB-C-002 | Support for the parm CONTEXT-ALLOW | 4.6.8 | O | N |
| ProvCont-CB-C-003 | Support for the parm PROXY-ID | 4.6.8 | O | Y<br>This value needs to be the same as that under PXLOGICAL |
| ProvCont-CB-C-004 | Support for parm NETWORK | 4.6.8 | O | N |
| ProvCont-CB-C-005 | Support for parm COUNTRY | 4.6.8 | O | N |
| ProvCont-CB-C-006 | Support for parm NAME | 4.6.8 | O | Y |

| Characteristic CLIENTIDENTITY | | | | |
|---|---|---|---|---|
| Item # | Function | Ref | Status | Motorola |
| ProvCont-CID-C-001 | Support for parm CLIENT-ID | 4.6.9 | O | N |

| Characteristic VENDORCONFIG | | | | |
|---|---|---|---|---|
| Item # | Function | Ref | Status | Motorola |
| ProvCont-CVC-C-001 | Support for parm NAME | 4.6.10 | O | N |
| ProvCont-CVC-C-002 | Support for other parameters than NAME | 4.6.10 | O | N |

| Characteristic APPLICATION | | | | |
|---|---|---|---|---|
| Item # | Function | Ref | Status | Motorola |
| ProvCont-CAP-C-001 | Support for parm APPID | 4.6.11 | O | Y<br>So far only for MMS, SyncML Data Synchronization & Email |
| ProvCont-CAP-C-002 | Support for parm PROVIDER-ID | 4.6.11 | O | Y<br>Limited use for Email |

| | | | | |
|---|---|---|---|---|
| ProvCont-CAP-C-003 | Support for parm NAME | 4.6.11 | O | Y |
| ProvCont-CAP-C-004 | Support for parm AACCEPT | 4.6.11 | O | N |
| ProvCont-CAP-C-005 | Support for parm APROTOCOL | 4.6.11 | O | N |
| ProvCont-CAP-C-006 | Support for parm TO-PROXY | 4.6.11 | O | N |
| ProvCont-CAP-C-007 | Support for parm TO-NAPID | 4.6.11 | O | Y<br>Limited use for Email |
| ProvCont-CAP-C-008 | Support for parm ADDR | 4.6.11 | O | Y<br>So far only for SyncML Data Synchronization & Email |

| Characteristic APPADDR | | | | |
|---|---|---|---|---|
| Item # | Function | Ref | Status | Motorola |
| ProvCont-CAA-C-001 | Support for parm ADDR | 4.6.12 | O | Y<br>Limited use for Email |
| ProvCont-CAA-C-002 | Support for parm ADDRTYPE | 4.6.12 | O | Y<br>Limited use for Email |

| Characteristic APPAUTH | | | | |
|---|---|---|---|---|
| Item # | Function | Ref | Status | Motorola |
| ProvCont-CAU-C-001 | Support for parm AAUTHLEVEL | 4.6.13 | O | Y<br>So far only for SyncML Data Synchronization |
| ProvCont-CAU-C-002 | Support for parm AAUTHTYPE | 4.6.13 | O | Y<br>So far only for SyncML Data Synchronization |
| ProvCont-CAU-C-003 | Support for parm AAUTHNAME | 4.6.13 | O | Y<br>So far only for SyncML Data Synchronization & Email |
| ProvCont-CAU-C-004 | Support for parm AAUTHSECRET | 4.6.13 | O | Y<br>So far only for SyncML |

| | | | | |
|---|---|---|---|---|
| | | | | Data Synchronization & Email |
| ProvCont-CAU-C-005 | Support for parm AAUTHDATA | 4.6.13 | O | N |

| Characteristic RESOURCE | | | | |
|---|---|---|---|---|
| Item # | Function | Ref | Status | Motorola |
| ProvCont-CRE-C-001 | Support for parm URI | 4.6.14 | O | Y<br>So far only for SyncML Data Synchronization |
| ProvCont-CRE-C-002 | Support for parm NAME | 4.6.14 | O | N |
| ProvCont-CRE-C-003 | Support for parm AACCEPT | 4.6.14 | O | Y<br>So far only for SyncML Data Synchronization |
| ProvCont-CRE-C-004 | Support for parm AAUTHTYPE | 4.6.14 | O | N |
| ProvCont-CRE-C-005 | Support for parm AAUTHNAME | 4.6.14 | O | N |
| ProvCont-CRE-C-006 | Support for parm AAUTHSECRET | 4.6.14 | O | N |
| ProvCont-CRE-C-007 | Support for parm AAUTHDATA | 4.6.14 | O | N |
| ProvCont-CRE-C-008 | Support for parm STARTPAGE | 4.6.14 | O | N |

| Characteristic ACCESS | | | | |
|---|---|---|---|---|
| Item # | Function | Ref | Status | Motorola |
| ProvCont-CAC-C-001 | Support for parm RULE | 4.6.15 | O | N |
| ProvCont-CAC-C-002 | Support for parm APPID | 4.6.15 | O | N |
| ProvCont-CAC-C-003 | Support for parm PORTNBR | 4.6.15 | O | N |

| | | | | |
|---|---|---|---|---|
| ProvCont-CAC-C-004 | Support for parm DOMAIN | 4.6.15 | O | N |
| ProvCont-CAC-C-005 | Support for parm TO-NAPID and/or parm TO-PROXY | 4.6.15 | O | N |
| ProvCont-CAC-C-006 | Support for parm TO-NAPID | 4.6.15 | O | N |
| ProvCont-CAC-C-007 | Support for parm TO-PROXY | 4.6.15 | O | N |

| Minimum Length of Parameter Fields | | | | |
|---|---|---|---|---|
| Item # | Function | Ref | Status | Motorola |
| ProvCont-MLP-C-001 | Support for minimum length of parm NAME | 5.1 | M | Y<br>For Email account Name is 15 |
| ProvCont-MLP-C-002 | Support for minimum length of parm NAP-ADDRESS | 5.1 | O | Y |
| ProvCont-MLP-C-003 | Support for minimum length of parm AUTHNAME | 5.1 | O | Y |
| ProvCont-MLP-C-004 | Support for minimum length of parm AUTHSECRET | 5.1 | O | Y |
| ProvCont-MLP-C-005 | Support for minimum length of parm PROXY-ID | 5.1 | O | Y |
| ProvCont-MLP-C-006 | Support for minimum length of parm DOMAIN | 5.1 | O | Y |
| ProvCont-MLP-C-007 | Support for minimum length of parm PROVURL | 5.1 | O | N |
| ProvCont-MLP-C-008 | Support for minimum length of parm PXAUTH-ID | 5.1 | O | N |
| ProvCont-MLP-C-009 | Support for minimum length of parm PXAUTH-PW | 5.1 | O | N |
| ProvCont-MLP-C-010 | Support for minimum length of parm STARTPAGE | 5.1 | M | Y |

| | | | | |
|---|---|---|---|---|
| ProvCont-MLP-C-011 | Support for minimum length of parm BASAUTH-ID | 5.1 | M | N |
| ProvCont-MLP-C-012 | Support for minimum length of parm BASAUTH-PW | 5.1 | M | N |
| ProvCont-MLP-C-013 | Support for minimum length of parm PXADDR | 5.1 | M | Y |
| ProvCont-MLP-C-014 | Support for minimum length of parm LINKSPEED | 5.1 | O | Y |
| ProvCont-MLP-C-015 | Support for minimum length of parm DNLINKSPEED | 5.1 | O | N |
| ProvCont-MLP-C-016 | Support for minimum length of parm LINGER | 5.1 | O | Y |
| ProvCont-MLP-C-017 | Support for minimum length of parm VALIDUNTIL | 5.1 | O | N |
| ProvCont-MLP-C-018 | Support for minimum length of parm PHYSICAL PROXY-ID | 5.1 | M | Y |
| ProvCont-MLP-C-019 | Support for minimum length of parm NAPID | 5.1 | M | Y |
| ProvCont-MLP-C-020 | Support for minimum length of parm CLIENT-ID | 5.1 | O | N |
| ProvCont-MLP-C-021 | Support for minimum length of parm PROXY-PROVIDER-ID | 5.1 | O | N |
| ProvCont-MLP-C-022 | Support for minimum length of parm PXADDR-FQDN | 5.1 | O | N |
| ProvCont-MLP-C-023 | Support for minimum length of parm PROXY-PW | 5.1 | O | N |
| ProvCont-MLP-C-024 | Support for minimum length of parm DNS-ADDR | 5.1 | O | Y |
| ProvCont-MLP-C-025 | Support for minimum length of parm APPID | 5.1 | O | Y |
| ProvCont-MLP-C-026 | Support for minimum length of parm PROVIDER-ID | 5.1 | O | Y |

| | | | | |
|---|---|---|---|---|
| ProvCont-MLP-C-027 | Support for minimum length of parm ADDR | 5.1 | O | Y |
| ProvCont-MLP-C-028 | Support for minimum length of parm APROTOCOL | 5.1 | O | N |
| ProvCont-MLP-C-029 | Support for minimum length of parm AAUTHNAME | 5.1 | O | Y |
| ProvCont-MLP-C-030 | Support for minimum length of parm AAUTHSECRET | 5.1 | O | Y |
| ProvCont-MLP-C-031 | Support for minimum length of parm AACCEPT | 5.1 | O | Y |
| ProvCont-MLP-C-032 | Support for minimum length of parm URI | 5.1 | O | Y |
| ProvCont-MLP-C-033 | Support for minimum length of parm REREG-THRESHOLD | 5.1 | O | N |
| ProvCont-MLP-C-034 | Support for minimum length of parm RULE | 5.1 | O | N |

# Appendix C: OTAP Enabled Phones

The OTAP feature was previously listed as a supported feature for MIB 1.2 and MIB 1.2.1. However, this is not correct. The feature is only supported in the products with the MIB version 2.0 or greater. A partial list of phones and the corresponding browser version is listed below. MIB version 2.2 and greater supports SIM Provisioning per the OMA SmartCard specification.

To obtain the browser version in any phone model, please refer to the "BrowserName" field in the User Agent Profile for the particular phone model. You can find this document posted at
http://motorola.handango.com/phoneconfig/<phone_model>/Profile/<phone_model>.rdf

(Note: replace <phone_model> with the model number, such as v600.)

MIB 2.0

C350 (April Release)
C350 (June/July Release)

MIB 2.1

A820
A835

MIB 2.2

E380
V300, V400, V500
E390
A845

MIB 2.2.1

V551
V3
E398
V80

E1000
V980
V975
C980
C975