SyncML Notification Initiated Session

Abstract

This document specifies the SyncML Device Management Notification Initiation package from the server. A management server can use this notification capability to cause the client to initiate a connection back to the management server.
SyncML Initiative

The following companies are Sponsors of the SyncML Initiative:

- Ericsson
- IBM
- Lotus
- Matsushita Communication Industrial Co., Ltd.
- Motorola
- Nokia
- Openwave
- Starfish Software
- Symbian

Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>V0.8</td>
<td>2001-11-21</td>
<td>Alpha release</td>
</tr>
<tr>
<td>V1.1</td>
<td>2002-02-15</td>
<td>First release</td>
</tr>
</tbody>
</table>
Copyright Notice

Copyright (c) Ericsson, IBM, Lotus, Matsushita Communication Industrial Co., Ltd., Motorola, Nokia, Openwave, Palm, Psion, Starfish Software, Symbian and others (2002). All Rights Reserved.

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

THIS DOCUMENT AND THE INFORMATION CONTAINED HEREIN ARE PROVIDED ON AN "AS IS" BASIS WITHOUT WARRANTY OF ANY KIND AND ERICSSON, IBM, LOTUS, MATSUSHITA COMMUNICATION INDUSTRIAL CO., LTD., MOTOROLA, NOKIA, OPENWAVE, PALM, PSION, STARFISH SOFTWARE, SYMBIAN AND ALL OTHER SYNCML SPONSORS DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL ERICSSON, IBM, LOTUS, MATSUSHITA COMMUNICATION INDUSTRIAL CO., LTD., MOTOROLA, NOKIA, OPENWAVE, PALM, PSION, STARFISH SOFTWARE, SYMBIAN OR ANY OTHER SYNCML SPONSOR BE LIABLE TO ANY PARTY FOR ANY LOSS OF PROFITS, LOSS OF BUSINESS, LOSS OF USE OF DATA, INTERRUPTION OF BUSINESS, OR FOR DIRECT, INDIRECT, SPECIAL OR EXEMPLARY, PUNITIVE OR CONSEQUENTIAL DAMAGES OF ANY KIND IN CONNECTION WITH THIS DOCUMENT OR THE INFORMATION CONTAINED HEREIN, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The above notice and this paragraph must be included on all copies of this document that are made.

Attention is called to the possibility that implementation of this specification may require use of subject matter covered by patent rights. By publication of this specification, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The SyncML Initiative is not responsible for identifying patents having necessary claims for which a license may be required by a SyncML Initiative specification or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

A patent/application owner has filed a statement of assurance that it will grant licenses under these rights without compensation or under reasonable rates and nondiscriminatory, reasonable terms and conditions to all applicants desiring to obtain such licenses. The SyncML Initiative makes no representation as to the reasonableness of rates and/or terms and conditions of the license agreements offered by patent/application owners. Further information may be obtained from the SyncML Initiative Executive Director.
Table of Contents

1 Introduction ........................................................................................................ 5
2 Formatting Conventions ....................................................................................... 5
    2.1 Message Sequence Chart Notation .............................................................. 5
3 Server Alerted Management session .................................................................... 5
4 Structure of General Notification Initiated Session Alert ............................... 6
    4.1.1 Syntax for the Initiation Notification ..................................................... 6
    4.1.2 Description of the fields ........................................................................ 7
5 SyncML Device Management Transport Dependant Profiles ......................... 9
    5.1 Package #0 delivered using WAP Push ...................................................... 10
        5.1.1 Using non WAP Push capable devices .............................................. 10
    5.2 Package #0 over OBEX ............................................................................ 10
6 References ........................................................................................................... 11
7 Appendix 1 - Example of Trigger Message from Server ............................... 12
1 Introduction

Many devices cannot continuously listen for connections from a management server. Other devices simply do not wish to “open a port” (i.e. accept connections) for security reasons. However, most devices can receive unsolicited messages, sometimes called “notifications”. Some handsets, for example, can receive SMS messages. Other devices may have the ability to receive other, similar datagram messages.

A management server can use this notification capability to cause the client to initiate a connection back to the management server. This connection might be over HTTP, WAP or another transport protocol.

The contents of such a “Notification Initiation Alert” might be empty, but the message itself may be signed such that the client can authenticate it. The result of receiving such an alert would be for the client to initiate a connection to the management server that sent the alert. In this scenario, the client might verify that this management server is among those authorized to request such activity. Alternatively, the contents of the alert might indicate that another management server should be contacted.

An identical effect of receiving a Notification Initiation Alert can also be caused in other ways. For example, the user interface (UI) of the device may allow the user to tell the client to initiate a management session. Or, the management client might initiate a session as the result of a timer expiring. Of course, a fault of some type in the device could also cause the management client to initiate a session.

2 Formatting Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY" and "OPTIONAL" in this document are to be interpreted as described in [1].

2.1 Message Sequence Chart Notation

Notation used in the message sequence charts (MSC):

Box – Indicates the start of a procedure or an internal process in a device

Hexagon – Indicates a condition that is needed to start the transaction below this hexagon

Arrow – Represents a message, or transaction

3 Server Alerted Management session

This notification message is intended to provide a possibility for the server to alert the client to perform a management session. When the server alerts the client, it can tell for example the protocol version, whether the server proposes the session to be a foreground or background event. It can also tell if the session is happening because server has some management objects to update or if the user caused the start of the session. The server MUST also send a digest that is included to prevent the Denial of Service (DoS) attacks.
Figure 1 describes the MSC how the server alerts management session.

![MSC diagram](image)

**Figure 1. MSC of the Server Alerted Management session**

The package flow presented above is one SyncML Device Management session that means that all messages have the same SyncML Session ID.

### 4 Structure of General Notification Initiated Session Alert

General Package#0 is the default format used for the Notification Initiated Session Trigger Message. This default format can be used if this document does not describe a special format for initialisation purposes.

The following figure describes the format of the General Package #0.

![General Package format](image)

**Figure 2. Format of the General Notification Trigger Message (Package#0)**

The MIME type for the General Notification Initiated Session Alert message is `application/vnd.syncml.notification` and the Content-Type code for that is `0x42`.

#### 4.1.1 Syntax for the Initiation Notification

The following ABNF [2] defines the syntax for the message. The order and the size of the fields MUST be same as specified in the following syntax of the Trigger Message.

```
<trigger-message> ::= <digest><trigger>
<digest> ::= 128*BIT ; 'MD5 Digest value'
```

---

Copyright © 2002 Ericsson, IBM, Lotus, Matsushita Communications Industrial Co., Ltd., Motorola, Nokia, Openwave, Palm, Psion, Starfish Software, Symbian, and others. All Rights Reserved.
<trigger> ::= <trigger-hdr><trigger-body>

<trigger-hdr> ::= <version><ui-mode><initiator><future-use>
<sessionid><length-identifier>

<version> ::= 10*BIT ; ‘Device Management Version’
<ui-mode> ::= <not-specified> / <background> / <informative> / <user-interaction> ; ‘Background/Informative/User Interaction session’
<not-specified> ::= “00” ; ‘2*bit value “0”’
<background> ::= “01” ; ‘2*bit value “1”’
<informative> ::= “10” ; ‘2*bit value “2”’
<user-interaction> ::= “11” ; ‘2*bit value “3”’
<initiator> ::= <client> / <server> ; ‘Server/User initiated’
<client> ::= “0” ; ‘1*bit value “0”’
<server> ::= “1” ; ‘1*bit value “1”’
<future-use> ::= 27*BIT ; ‘Reserved for future DM use’
<sessionid> ::= 16*BIT ; ‘Session identifier’
<length-identifier> ::= 8*BIT ; ‘Server Identifier length’

<trigger-body> ::= <server-identifier>[<vendor-specific>]

<server-identifier> ::= <length-source>*CHAR ; ‘Server Identifier’
/vendor-specific> ::= n*BIT ; ‘Optional vendor specific info’

4.1.2 Description of the fields

4.1.2.1 Trigger Message
The <trigger-message> field specifies the message causing the client to connect to the server.

4.1.2.2 Digest
The <digest> field specifies the MD5 Digest authentication. The Digest is computed as
Digest = H(B64(H(server-identifier:password)):nonce:B64(H(trigger))). Length of MD5
Digest is 128 bits.

4.1.2.3 Trigger
The <trigger> field is container for the trigger-hdr and trigger-body fields.

4.1.2.4 Header of the Trigger Message
The <trigger-hdr> field specifies the header of the Trigger Message.

4.1.2.5 Body of the Trigger Message
The <trigger-body> field specifies the body of the Trigger Message.
4.1.2.6 Version information

The `<version>` field specifies the version of the SyncML Device Management Package #0 and the SyncML DM protocol used in the SyncML DM server. This value is specified by using the 10 bits in the Trigger Message. The supported version is counted as Supported version = \( \text{DEC} \left( \frac{\text{version}}{10} \right) \). So, first the bit value is transferred to the numeric and then divided by ten. So basically the biggest possible version is ‘102.3’ and the version ‘1.0’ is specified as ‘0000001010’.

The backward compatibility requirements for protocol versions are specified in more detail in the SyncML Interoperability Testing Process [3].

4.1.2.7 User Interaction mode

The `<ui-mode>` field specifies the server recommendations whether the server wants the management session to be executed in background or show a notification to the user. A client SHOULD follow this recommendation.

The values the User Interaction mode can have:

- Not specified - The `<not-specified>` field in `<user-interaction>` field specifies that the server don’t have a recommendation to this element. This value is specified by using the 2 bits and the bit value for not specified action is “00”.

- Background management action - The `<background>` field specifies that the server recommends the management action SHOULD be done as a background event. This value is specified by using the 2 bits and the bit value for background action is “01”.

- Informative management action - The `<informative>` field specifies that the server recommends the client to display an informative notification or maybe emitting a beep sound announcing the beginning of the provisioning session to the device user. This value is specified by using the 2 bits and the bit value for informative notification is “10”.

- User Interaction before the management action - The `<user-interaction>` field specifies that the server recommend the client to prompt the device user for acceptance of the offered management session before the management session takes place. This value is specified by using the 2 bits and the bit value for user displayable notification is “11”.

4.1.2.8 Initiator of the Management action

The `<initiator>` field specifies the server recommendations whether the management action is initiated because the end user requested it or because the server has management objects to manage. A client SHOULD follow this recommendation.

The values the Initiator of the Management action can have:
• Client (End User) Initiated management action - The <client> field specifies that the end user caused the device management session to start. This value is specified by using 1 bit and the bit value for end user initiated management session is "0".

• Server Initiated management action - The <server> field specifies that the server (operator, enterprise) caused the device management session to start. This value is specified by using 1 bit and the bit value for Server initiated management session is "1".

The <client> and <server> values do not convey any information related to "sync type" (SyncML Synchronization protocol document [4] for details of Sync Types).

4.1.2.9 Future use of the Device Management

The <future-use> field is reserved for the future fields for SyncML Device Management. The reserved space is 27 bits long and the bit value for bits not yet in use MUST be “0”.

4.1.2.10 Session identifier

The <sessionid> field specifies the identifier of the SyncML DM session associated with the SyncML Message. This value is specified by using the 16 bits in the Trigger Message. The Session ID MUST be different between different management session Trigger Messages and the Client MUST use this Session ID when it connects to the SyncML DM Server. If the server triggers the same management session several times, it is recommended the same Session ID be used.

4.1.2.11 Length of the Identifier

The <length-identifier> field specifies the length of the Server Identifier of the management server. The value of the Length Identifier is counted as Length of the server-identifier = DEC (length-identifier).

4.1.2.12 Server Identifier

The <server-identifier> field specifies the Server Identifier of the management server. Length of source is specified in the <length-source> field.

4.1.2.13 Vendor specific information

The optional <vendor specific> field is used to specify vendor specific information. This field is after the source field and the rest of the Trigger Message size can be fulfilled by the vendor specific information.

5 SyncML Device Management Transport Dependant Profiles

The following sections illustrate the transport dependant profiles for sending a trigger from SyncML Device Management Server to a SyncML Device Management Client.
5.1 Package #0 delivered using WAP Push

The WAP Push framework provides a means for a Push Initiator (PI) to send information to a mobile terminal via a Push Proxy Gateway (PPG) in an asynchronous manner (see [5] for an overview). It is assumed that the SyncML DM server will act as a PI, but it is also possible for the server to communicate directly with the mobile terminal if it is able to operate as a PPG.

When the WAP Push framework is used to deliver Package #0, the non-secure connectionless WSP [8] session service is utilised as defined in [6]. The following rules MUST be adhered to as well as the order of the WSP headers:

- The Content-Type header [7] MUST include the MIME media type for Packet #0 as defined in [9]. The Content-Type code 0x42 registered with WINA [10] MUST be used instead of the textual representation of the MIME code.

- The X-WAP-Application-ID header [7] MUST include the application-id associated with the Sync ML Device Management User Agent (registered with WINA [10]). The application-id code 0x07 MUST be used instead of the textual representation of the Application-id.

- Other headers may be included if it is known that the SyncML DM Client can interpret them in a useful manner. However, it must be ensured that the total length of the WDP and WSP headers never exceeds 48 bytes to ensure that there is sufficient space for the payload.

- The push message is sent to the default non-secure connectionless push port (2948)

The message payload has been designed to fit into a single short message when SMS is used to deliver WAP Push. If the WAP Push message does not fit into a single SMS message the concatenated messages MUST be used.

5.1.1 Using non WAP Push capable devices

If the receiver is not a WAP device, it is very unlikely that any other application would be active on the same port, which has been publicly registered with IANA. The decoding of the message headers is very straightforward even if the device lacks a full WAP stack and therefore the device MUST examine if the message has been sent to the WAP push port (2948) and if the Application-ID and the MIME type are one assigned to the SyncML DM Notification Initiation Package. If this information is correct then the message MUST be routed to the SyncML Device Management application.

5.2 Package #0 over OBEX

Local Notification Initiated Session over OBEX is done inside the PUT command of the OBEX protocol. This happens in the same way as sending the SyncML messages over OBEX to a SyncML client (See the SyncML OBEX Binding specification [11]).
6 References

[1] Key words for use in RFCs to Indicate Requirement Levels, RFC 2119, IETF.
[9] Internet Assigned Numbers Authority, IANA.
[10] WAP Interim Naming Authority, WINA.
## 7 Appendix 1 - Example of Trigger Message from Server

Example WAP Push over SMS containing the trigger information:

<table>
<thead>
<tr>
<th>Binary value</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>06</td>
<td>User-Data-Header (UDHL) Length = 6 bytes</td>
<td>WDP layer (start WDP headers).</td>
</tr>
<tr>
<td>05</td>
<td>UDH IE identifier: Port numbers</td>
<td></td>
</tr>
<tr>
<td>04</td>
<td>UDH port number IE length</td>
<td></td>
</tr>
<tr>
<td>0B</td>
<td>Destination port (high)</td>
<td>Port number 2948</td>
</tr>
<tr>
<td>84</td>
<td>Destinating port (low)</td>
<td></td>
</tr>
<tr>
<td>C0</td>
<td>Originating port (high)</td>
<td>Port number chosen by sender</td>
</tr>
<tr>
<td>02</td>
<td>Originating port (low)</td>
<td>WDP layer (end WDP headers)</td>
</tr>
<tr>
<td>01</td>
<td>Transaction ID / Push ID</td>
<td>WSP layer (start WSP headers)</td>
</tr>
<tr>
<td>06</td>
<td>PDU type (push)</td>
<td></td>
</tr>
<tr>
<td>03</td>
<td>Headerslength (content type+headers)</td>
<td></td>
</tr>
<tr>
<td>C2</td>
<td>Content type code</td>
<td>MIME-Type</td>
</tr>
<tr>
<td>AF</td>
<td>X-WAP-Application-ID</td>
<td></td>
</tr>
<tr>
<td>87</td>
<td>Id for urn: x-wap-application:syncml.dm</td>
<td>WSP layer (end WSP headers)</td>
</tr>
<tr>
<td>128-bit digest value</td>
<td>Digest</td>
<td></td>
</tr>
<tr>
<td>Binary ‘0000001011’</td>
<td>Version ‘1.1’</td>
<td></td>
</tr>
<tr>
<td>Binary ‘01’</td>
<td>UI-Mode ‘1’</td>
<td></td>
</tr>
<tr>
<td>Binary ‘00’</td>
<td>Initiator ‘0’</td>
<td></td>
</tr>
<tr>
<td>Binary ‘00000000000000000000000000000000’</td>
<td>Future DM use</td>
<td></td>
</tr>
<tr>
<td>Binary ‘000000000000000001’</td>
<td>SessionID ‘1’</td>
<td></td>
</tr>
<tr>
<td>Binary ‘0000100010’</td>
<td>Server ID length ‘34’</td>
<td></td>
</tr>
<tr>
<td>68, 74, 74, 70, 3A, 2F, 2F, 77, 77, 77, 2E, 6D, 6E, 67, 6D, 74, 73, 65, 72, 76, 65, 72, 2E, 63, 6F, 6D, 2F, 6D, 61, 6E, 61, 67, 65, 2F</td>
<td>String &quot;<a href="http://www.mngmtserver.com/manage/">http://www.mngmtserver.com/manage/</a>&quot;</td>
<td>Server Identifier</td>
</tr>
</tbody>
</table>