



SyncML Device Information, version 1.1.2

Approved Version 12-June-2003

Open Mobile Alliance
OMA-SyncML-DevInfo_V1_1_2-20030612-A

Continues the Technical Activities
Originated in the SyncML Initiative



Use of this document is subject to all of the terms and conditions of the Use Agreement located at <http://www.openmobilealliance.org/UseAgreement.html>.

Unless this document is clearly designated as an approved specification, this document is a work in process, is not an approved Open Mobile Alliance™ specification, and is subject to revision or removal without notice.

You may use this document or any part of the document for internal or educational purposes only, provided you do not modify, edit or take out of context the information in this document in any manner. Information contained in this document may be used, at your sole risk, for any purposes. You may not use this document in any other manner without the prior written permission of the Open Mobile Alliance. The Open Mobile Alliance authorizes you to copy this document, provided that you retain all copyright and other proprietary notices contained in the original materials on any copies of the materials and that you comply strictly with these terms. This copyright permission does not constitute an endorsement of the products or services. The Open Mobile Alliance assumes no responsibility for errors or omissions in this document.

Each Open Mobile Alliance member has agreed to use reasonable endeavors to inform the Open Mobile Alliance in a timely manner of Essential IPR as it becomes aware that the Essential IPR is related to the prepared or published specification. However, the members do not have an obligation to conduct IPR searches. The declared Essential IPR is publicly available to members and non-members of the Open Mobile Alliance and may be found on the “OMA IPR Declarations” list at <http://www.openmobilealliance.org/ipr.html>. The Open Mobile Alliance has not conducted an independent IPR review of this document and the information contained herein, and makes no representations or warranties regarding third party IPR, including without limitation patents, copyrights or trade secret rights. This document may contain inventions for which you must obtain licenses from third parties before making, using or selling the inventions. Defined terms above are set forth in the schedule to the Open Mobile Alliance Application Form.

NO REPRESENTATIONS OR WARRANTIES (WHETHER EXPRESS OR IMPLIED) ARE MADE BY THE OPEN MOBILE ALLIANCE OR ANY OPEN MOBILE ALLIANCE MEMBER OR ITS AFFILIATES REGARDING ANY OF THE IPR'S REPRESENTED ON THE “OMA IPR DECLARATIONS” LIST, INCLUDING, BUT NOT LIMITED TO THE ACCURACY, COMPLETENESS, VALIDITY OR RELEVANCE OF THE INFORMATION OR WHETHER OR NOT SUCH RIGHTS ARE ESSENTIAL OR NON-ESSENTIAL.

THE OPEN MOBILE ALLIANCE IS NOT LIABLE FOR AND HEREBY DISCLAIMS ANY DIRECT, INDIRECT, PUNITIVE, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR EXEMPLARY DAMAGES ARISING OUT OF OR IN CONNECTION WITH THE USE OF DOCUMENTS AND THE INFORMATION CONTAINED IN THE DOCUMENTS.

© 2003 Open Mobile Alliance Ltd. All Rights Reserved.

Used with the permission of the Open Mobile Alliance Ltd. under the terms set forth above.

Contents

1. SCOPE	5
2. REFERENCES.....	6
2.1 NORMATIVE REFERENCES	6
2.2 INFORMATIVE REFERENCES	6
3. TERMINOLOGY AND CONVENTIONS	7
3.1 CONVENTIONS	7
3.2 DEFINITIONS.....	7
3.3 ABBREVIATIONS	8
4. INTRODUCTION.....	9
5. DEVICE INFORMATION	10
5.1 XML USAGE.....	10
5.2 MIME USAGE	10
5.3 DEVICE INFORMATION ELEMENT DESCRIPTIONS	11
5.3.1 CTCap	11
5.3.2 CType	12
5.3.3 DataStore.....	12
5.3.4 DataType.....	13
5.3.5 DevID.....	14
5.3.6 DevInf	15
5.3.7 DevTyp.....	15
5.3.8 DisplayName.....	16
5.3.9 DSMem	16
5.3.10 Ext	17
5.3.11 FwV.....	17
5.3.12 HwV.....	17
5.3.13 Man	18
5.3.14 MaxGUIDSize.....	18
5.3.15 MaxID	18
5.3.16 MaxMem.....	19
5.3.17 Mod.....	19
5.3.18 OEM.....	19
5.3.19 ParamName.....	20
5.3.20 PropName.....	21
5.3.21 Rx	23
5.3.22 Rx-Pref.....	23
5.3.23 SharedMem	24
5.3.24 Size.....	24
5.3.25 SourceRef.....	25
5.3.26 SupportLargeObjs	25
5.3.27 SupportNumberOfChanges	25
5.3.28 SwV.....	26
5.3.29 SyncCap	26
5.3.30 SyncType	27
5.3.31 Tx	27
5.3.32 Tx-Pref	28
5.3.33 UTC.....	28
5.3.34 ValEnum	29
5.3.35 VerCT.....	31
5.3.36 VerDTD	31
5.3.37 XNam	31
5.3.38 XVal.....	32
6. DEVICE INFORMATION DTD	33

7. WBXML DEFINITIONS	35
8. EXAMPLES	37
9. MIME MEDIA TYPE REGISTRATION.....	40
9.1 APPLICATION/VND.SYNCML-DEVINF+XML	40
9.2 APPLICATION/VND.SYNCML-DEVINF+WBXML	41
APPENDIX A. STATIC CONFORMANCE REQUIREMENTS (NORMATIVE)	43
APPENDIX B. CHANGE HISTORY (INFORMATIVE)	45

1. Scope

The SyncML Initiative, Ltd. was a not-for-profit corporation formed by a group of companies who co-operated to produce an open specification for data synchronization. Prior to SyncML, data synchronization and device management had been based on a set of different, proprietary protocols, each functioning only with a very limited number of devices, systems and data types. These non-interoperable technologies have complicated the tasks of users, manufacturers, service providers, and developers. Further, a proliferation of different, proprietary data synchronization protocols has placed barriers to the extended use of mobile devices, has restricted data access and delivery and limited the mobility of the users.

SyncML Components

SyncML is a data synchronization specification that contains the following main components:

- An XML-based representation protocol
- A synchronization protocol
- Transport bindings for the synchronization protocol

The data representation specifies an XML DTD that allows the representation of all the information required to perform synchronization, including data, metadata and commands. The synchronization protocols specifies how SyncML messages conforming to the DTD are exchanged in order to allow a SyncML client and server to exchange additions, deletes, updates and other status information.

The synchronization protocol supports both two-way and one-way synchronization.

There are also DTDs which define the representation of information about the device such as memory capacity, and the representation of various types of meta information such as security credentials.

Although the SyncML specification defines transport bindings that specify how to use a particular transport to exchange messages and responses, the SyncML representation, and synchronization protocols are transport-independent. Each SyncML package is completely self-contained, and could in principle be carried by any transport. The initial bindings specified are HTTP, WSP and OBEX, but there is no reason why SyncML could not be implemented using email or message queues, to list only two alternatives. Because SyncML messages are self-contained, multiple transports may be used without either the server or client devices having to be aware of the network topology. Thus, a short-range OBEX connection could be used for local connectivity, with the messages being passed on via HTTP to an Internet-hosted synchronization server.

Either the client or the server may initiate a synchronization session, and both one and two-way synchronization are supported. Both linear (point-to-point) and star (one-to-many) synchronization topologies may be implemented using SyncML.

To reduce the data size, a binary coding of SyncML based on the WAP Forum's WBXML is defined. Messages may also be passed in clear text if required. In this and other ways SyncML addresses the bandwidth and resource limitations imposed by mobile devices.

SyncML is both data type and data store independent. SyncML can carry any data type which can be represented as a MIME object. To promote interoperability between different implementations of SyncML, the specification includes the representation formats used for common PIM data

This document specifies the device information syntax and semantics used by the SyncML data synchronization protocol.

2. References

2.1 Normative References

- [DEIF] “Data elements and interchange formats - Information interchange - Representation of dates and times”, [URL:http://www.iso.ch/iso/en/ISOOnline.frontpage](http://www.iso.ch/iso/en/ISOOnline.frontpage)
- [DSREPU] “SyncML Representation Protocol, Data Synchronization Usage”, Open Mobile Alliance™, OMA-SyncML-DataSyncRep-V1_1_2, [URL:http://www.openmobilealliance.org/tech/docs](http://www.openmobilealliance.org/tech/docs)
- [SYNCMETA] “SyncML Meta Information”, Open Mobile Alliance™, OMA-SyncML-MetaInformation-V1_1_2, [URL:http://www.openmobilealliance.org/tech/docs](http://www.openmobilealliance.org/tech/docs)
- [RFC2045] “Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies”, N. Freed & N. Borenstein, November 1996, [URL:http://www.ietf.org/rfc/rfc2045.txt](http://www.ietf.org/rfc/rfc2045.txt)
- [RFC2199] “Key words for use in RFCs to Indicate Requirement Levels”, S. Bradner, March 1997, [URL:http://www.ietf.org/rfc/rfc2119.txt](http://www.ietf.org/rfc/rfc2119.txt)
- [SYNCPRO] “SyncML Synchronization Protocol”, Open Mobile Alliance™, OMA-SyncML-DataSyncProtocol-V1_1_2, [URL:http://www.openmobilealliance.org/tech/docs](http://www.openmobilealliance.org/tech/docs).
- [WBXML] “WAP Binary XML Content Format Specification”, WAP Forum™, WAP-240-WBXML, [URL:http://www.openmobilealliance.org/tech/docs](http://www.openmobilealliance.org/tech/docs)

2.2 Informative References

None.

3. Terminology and Conventions

3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [[RFC2199]].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

Any reference to components of the SyncML DTD or XML snippets is specified in this `typeface`.

3.2 Definitions

Application - A SyncML application that supports the SyncML protocol. The application can either be the originator or recipient of the SyncML protocol commands. The application can act as a SyncML client or a SyncML server.

Capabilities exchange - The SyncML capability that allows a client and server to exchange what device, user and application features they each support.

Client - A SyncML Client refers to the protocol role when the application issues SyncML "request" messages. For example in data synchronization, the Sync SyncML Command in a SyncML Message.

Command - A SyncML Command is a protocol primitive. Each SyncML Command specifies to a recipient an individual operation that is to be performed. For example, the SyncML Commands supported by this specification include Add, Alert, Atomic, Copy, Delete, Exec, Get, Map, Replace, Search, Sequence and Sync.

Data - A unit of information exchange, encoded for transmission over a network.

Data exchange - The act of sending, requesting or receiving a set of data elements.

Data format - The encoding used to format a data type. For example, characters or integers or character encoded binary data.

Data type - The schema used to represent a data object (e.g., text/calendar MIME content type for an iCalendar representation of calendar information or text/directory MIME content type for a vCard representation of contact information).

Device Information - A document or object store (i.e., a database) on the source device that records information about the capabilities of the source device.

GUID (Global Unique Identifier) – A number assigned to an object in a database. GUID values are never reused. Note that in practice, numbers do not have to be unique forever, they MUST only be unique as long as they exist in some mapping table.

Message - A SyncML Message is the primary contents of a SyncML Package. It contains the SyncML Commands, as well as the related data and meta-information. The SyncML Message is an XML document.

Originator - The network device that creates a SyncML request.

Package - A SyncML Package is the complete set of commands and related data elements that are transferred between an originator and a recipient. The SyncML package can consist of one or more SyncML Messages.

Parser - Refers to an XML parser. An XML parser is not absolutely required to support SyncML. However, a SyncML implementation that integrates an XML parser may be easier to enhance.

This document assumes that the reader has some familiarity with XML syntax and terminology.

Representation protocol - A well-defined format for exchanging a particular form of information. SyncML is a representation protocol for conveying data synchronization and device management operations.

Request – A message or a command sent from a device to another.

Server - A SyncML Server refers to the protocol role when an application issues SyncML "response" messages. For example in the case of data synchronization, a Results Command in a SyncML Message.

Temporary GUID – A temporary number assigned by the server to an object in a database (See also GUID.). Temporary GUID values are valid till the map operation for the items, with which the temporary GUIDs are associated, has been received from the client. After that the temporary GUID can be erased.

3.3 Abbreviations

4. Introduction

This document defines the Document Type Definition (DTD) for the XML representation of the Device Information object (DevInf.DTD). The DevInf.DTD is intended to be used to exchange device specific information. Exchange of device specific information such as available memory and item identifiers, supported local databases is a prerequisite to successful data synchronization.

Data synchronization provides the means for two different networked object stores to remain in identical states. Different forms of data synchronization can be categorized into one of a number of topologies, based on the architecture used by a data synchronization server, or sync engine. Sync engines need to understand the features of a device they synchronize with. This information is often stored in a Device Information document on the target device.

5. Device Information

5.1 XML Usage

The device information is represented in a mark up language defined by [WBXML]. The Device Information DTD (Document Type Definition) defines the XML document type used to represent information about the capabilities of a data synchronization client device.

Device Information documents are specified using well-formed XML. However, they need not be valid XML. That is, the Device Information documents do not need to specify the XML prolog. They only need to specify the body of the XML document. This restriction allows for Device Information documents to be specified with greater terseness than well-formed, valid XML documents.

The Device Information DTD makes use of XML name spaces. Name spaces must be declared on the first element type that uses an element type from the name space.

Names in XML are case sensitive. By convention in the Device Information DTD, the element type and attribute list names are specified with a "Hungarian" like notation of the first character in each word of the name in upper case text and remainder of the characters in each word of the names specified in lower case text. For example, `DevInf` for the Device Information root element type tag.

The element types in the Device Information DTD are defined within a namespace defined in section 6.

The formal public identifier (FPI) is the traditional format for specifying unique identifiers for XML entities. The FPI for the DTD described in this specification is:

```
-//SyncML//DTD DevInf 1.1//EN
```

The name for the file object corresponding to this document on a device MUST be:

```
devinf11
```

The Device Information DTD also makes use of XML standard attributes, such as `xml:lang`. Any XML standard attribute can be used in a SyncML document.

XML can be viewed as more verbose than alternative binary representations. This is often cited as a reason why it may not be appropriate for low bandwidth network protocols. In most cases, this specification uses shortened element type and attribute names. This provides a minor reduction in verbosity. Additionally, the Device Information documents can be encoded in a tokenized, binary format defined by [WBXML]. The token values used to encode the Device Information documents are defined in chapter 7 of this document. The use of [WBXML] format is external to this specification and should be transparent to any XML application supporting this DTD. The combination of the use of shortened element type and attribute names and an alternative binary format makes this specification competitive, from a compressed format perspective, with alternative, but private, binary representations for Device Information documents.

One of the main advantages of XML is that it is a widely accepted International recommendation for text document markup. It provides for both human readability and machine processability. In addition, XML allows the originator to capture the structure of a document, not just it's content. This is extremely useful for applications such as data synchronization, where not just content, but structure semantics is often exchanged.

The SyncML Device Information document also can be identified as a MIME content type. MIME is the Internet standard for identifying multipurpose message contents. It provides a useful mechanism for differentiating between different content and document types.

5.2 MIME Usage

The [RFC2045] Internet standard provides an industry-accepted mechanism for identifying different content types. A MIME media type identifies the SyncML Device Information document. The media type for the Device Information

document is registered within the vendor tree. There are two MIME content types for the Device Information document. The MIME content type of `application/vnd.syncml-devinf+xml` identifies the clear-text XML representation for the Device Information document. The MIME content type of `application/vnd.syncml-devinf+wbxml` identifies the WBXML binary representation for the Device Information document. Section 9 of this specification specifies the MIME content type registration for these two MIME media types.

One of these two MIME content types **MUST BE** used for identifying Device Information documents within transport and session level protocols that support MIME content types.

5.3 Device Information Element Descriptions

The following element types are included in the Device Information DTD.

5.3.1 CTCap

Usage: Specifies the content type capabilities of the device.

Parent Elements: `DevInf`

Restrictions: The content type capabilities of the device **SHOULD** be defined.

Content Model:

```
CTCap ((CTType, (PropName, (ValEnum+ | (DataType, Size?))?, DisplayName?,
(ParamName, (ValEnum+ | (DataType, Size?))?, DisplayName?)*))+)
```

Attributes: None.

Example:

```
<CTCap>
  <CTType>text/x-vcard</CTType>
  <PropName>BEGIN</PropName>
    <ValEnum>VCARD</ValEnum>
  <PropName>END</PropName>
    <ValEnum>VCARD</ValEnum>
  <PropName>VERSION</PropName>
    <ValEnum>2.1</ValEnum>
  <PropName>FN</PropName>
    <DataType>chr</DataType>
  <PropName>N</PropName>
  <PropName>TEL</PropName>
    <ParamName>WORK</ParamName>
    <ParamName>HOME</ParamName>
  <CTType>text/x-vcalendar</CTType>
```

```

<PropName>BEGIN</PropName>
  <ValEnum>VCALENDAR</ValEnum>
  <ValEnum>VEVENT</ValEnum>
<PropName>VERSION</PropName>
  <ValEnum>1.0</ValEnum>
<PropName>DTSTART</PropName>
  <DataType>datetime</DataType>
<PropName>DTEND</PropName>
  <DataType>datetime</DataType>
<PropName>DESCRIPTION</PropName>
</CTCap>

```

5.3.2 CType

Usage: Specifies the type of a supported content type.

Parent Elements: CTCap, Rx, Rx-Pref, Tx, Tx-Pref

Restrictions: If a parent element is present, this element type is required. The value for this element can be e.g., text/x-vcard, text/vcard, text/x-vcalendar, or text/calendar. Some other possible values for this element are specified in the SyncML representation protocol [DSREPU]. Other values can also be specified.

Content Model:

```
(#PCDATA)
```

Attributes: None.

Example:

```

<CTCap>
  <CTType>text/vcard</CTType>
</CTCap>

```

5.3.3 DataStore

Usage: Specifies the properties of a given local datastore.

Parent Elements: DevInf

Restrictions: One or more of the element types are required. One element type is required for each of the local datastores.

Content Model:

```
(SourceRef, DisplayName?, MaxGUIDSize?, Rx-Pref, Rx*, Tx-Pref, Tx*,
DSMem?, SyncCap)
```

Attributes: None.

Example:

```
<DataStore>
  <SourceRef>./contacts</SourceRef>
  <DisplayName>Addressbook</DisplayName>
  <MaxGUIDSize>32</MaxGUIDSize>
  <Rx-Pref>
    <CTType>text/vcard</CTType>
    <VerCT>3.0</VerCT>
  </Rx-Pref>
  <Rx>
    <CTType>text/x-vcard</CTType>
    <VerCT>2.1</VerCT>
  </Rx>
  <Tx-Pref>
    <CTType>text/vcard</CTType>
    <VerCT>3.0</VerCT>
  </Tx-Pref>
  <Tx>
    <CTType>text/x-vcard</CTType>
    <VerCT>2.1</VerCT>
  </Tx>
  <DSMem>
    <SharedMem/>
    <MaxMem>65539</MaxMem>
    <MaxID>512</MaxID>
  </DSMem>
  <SyncCap>
    <SyncType>1</SyncType>
    <SyncType>7</SyncType>
  </SyncCap>
</DataStore>
```

5.3.4 DataType

Usage: Specifies the datatype of a given content type property or parameter.

Parent Elements: CTCap

Restrictions: Type values for this element type can be one of chr, int, bool, bin, datetime, phonenum as defined in the table below. Other values can also be specified.

Data Type	Datatype of a content type property or parameter
chr	Character
int	Integer
bool	Boolean
bin	Binary
datetime	Date and time of day
phonenum	Phone number

Content Model:

(#PCDATA)

Attributes: None.

Example:

```
<PropName>FN</PropName>
  <DataType>chr</DataType>
<PropName>TEL</PropName>
  <ParamName>WORK</ParamName>
    <DataType>phonenum</DataType>
  <ParamName>HOME</ParamName>
    <DataType>phonenum</DataType>
```

5.3.5 DevID

Usage: Specifies the identifier of the source synchronization device.

Parent Elements: DevInf

Restrictions: The content information MUST specify a theoretically, globally unique identifier. This element type is mandatory.

Content Model:

(#PCDATA)

Attributes: None.

Example:

```
<DevID>1218182THD012345-2</DevID>
```

5.3.6 DevInf

Usage: Specifies the root or document element type of the Device Information document.

Parent Elements: None.

Restrictions: This element type is mandatory and MUST be the root or document element.

Content Model:

```
(VerDTD, Man?, Mod?, OEM?, FwV?, SwV?, HwV?, DevID, DevTyp, UTC?, SupportLargeObjs?, SupportNumberOfChanges?, DataStore+, CTCap*, Ext*)
```

Attributes:

Name	Type	Occurrence	Description
xmlns	CDATA	IMPLICIT	Must be present with a value of 'syncml:devinf'.

Example:

```
<DevInf>
  <VerDTD>1.1</VerDTD>
  <DevID>1218182THD012345-2</DevID>
  <DevTyp>pager</DevTyp>
  <DataStore>
    ...
  </DataStore>
</DevInf>
```

5.3.7 DevTyp

Usage: Specifies the type of the source synchronization device.

Parent Elements: DevInf

Restrictions: Type values for this element type can be e.g. pager, handheld, pda, phone, smartphone, server, workstation, as defined in the table below. Other values can also be specified. This element type is mandatory.

DevTyp	Type of the device
pager	Pager
handheld	Handheld PC/PDA
pda	Palm sized PC/PDA
phone	Cellular phone

smartphone	Smartphone
server	Server-class computer
workstation	Workstation-class computer

Content Model:

```
(#PCDATA)
```

Attributes: None.**Example:**

```
<DevTyp>pager</DevTyp>
```

5.3.8 DisplayName

Usage: Specifies the display name of a given local datastore, or the display name of a given content type property or parameter.

Parent Elements: DataStore, CTCap

Restrictions: This element type is optional.

Content Model:

```
(#PCDATA)
```

Attributes: None.

Example: The following example specifies the display name of the contacts datastore.

```
<DataStore>
  <SourceRef>./contacts</SourceRef>
  <DisplayName>Addressbook</DisplayName>
</DataStore>
```

5.3.9 DSMem

Usage: Specifies the maximum memory and item identifier for a given local datastore.

Parent Elements: DataStore

Restrictions: The element type is optional.

Content Model:

```
(SharedMem?, MaxMem?, MaxID?)
```

Attributes: None.

Example: The following example specifies a shared datastore memory.

```
<DSMem>
  <SharedMem/>
```



```
<MaxMem>65539</MaxMem>
<MaxID>512</MaxID>
</DSMem>
```

5.3.10 Ext

Usage: Specifies the non-standard, experimental extensions supported by the device. The extensions are specified in terms of the XML element type name and the value.

Parent Elements: DevInf

Restrictions: The Ext element type MUST specify the extension element name. It may also specify one or more enumerated values. Multiple non-standard extensions can be specified by specifying the Ext element type multiple times. This element type is optional.

Content Model:

```
(XNam, Xval*)
```

Attributes: None.

Example: The following example specifies a non-standard extension, named "CliVer" for a fictitious company, Foo, which takes values of "5.0", "5.01" or "5.02".

```
<Ext><XNam>X-Foo-CliVer</XNam><XVal>5.0</XVal><XVal>5.01</XVal>
<XVal>5.02<XVal></Ext>
```

5.3.11 FwV

Usage: Specifies the firmware version of the device.

Parent Elements: DevInf

Restrictions: If there is no firmware version of the device available, then the content information can also be a date. If the content information is a date, then it MUST be formatted as a complete representation, basic format of an [DEIF] date or date and UTC time of day. For example, 19980114 or 19990714T133000Z. Only hours, minutes and second MUST be specified in the time component. This element type is optional.

Content Model:

```
(#PCDATA)
```

Attributes: None.

Example:

```
<FwV>1.01</FwV>
```

5.3.12 HwV

Usage: Specifies the hardware version of the device.

Parent Elements: DevInf

Restrictions: If there is no hardware version of the device available, then the content information can also be a date. If the content information is a date, then it MUST be formatted as a complete representation, basic format of an [DEIF] date or date and UTC time of day. For example, 19980114 or 19990714T133000Z. Only hours, minutes and second MUST be specified in the time component. This element type is optional.

Content Model:

```
(#PCDATA)
```

Attributes: None.**Example:**

```
<HwV>0.1a</HwV>
```

5.3.13 Man

Usage: Specifies the name of the manufacturer of the device.**Parent Elements:** DevInf**Restrictions:** This element type is optional.**Content Model:**

```
(#PCDATA)
```

Attributes: None.**Example:**

```
<Man>Foo Industries, Inc.</Man>
```

5.3.14 MaxGUIDSize

Usage: Specifies the maximum size of a global unique identifier for a given local datastore, in bytes the device is able to receive and store.**Parent Elements:** DataStore**Restrictions:** Content information **MUST** be specified as the decimal integer number indicating the maximum size, in bytes of the temporary GUID the client device is able to receive and store for a given local datastore, and the server is allowed to send. The device acting as a client **MUST**, and the device acting as a server **MUST NOT** send this information.**Content Model:**

```
(#PCDATA)
```

Attributes: None.**Example:** The following is an example of a client capable of receiving and storing a maximum 2 bytes long GUID.

```
<MaxGUIDSize>2</MaxGUIDSize>
```

5.3.15 MaxID

Usage: Specifies the maximum number of items that can be stored in a given local datastore.**Parent Elements:** DSMem**Restrictions:** Content information **MUST** be specified as the decimal integer number of maximum item identifiers that are available for all items in the local datastore. This element type is optional.

Content Model:

```
(#PCDATA)
```

Attributes: None.**Example:** The following is an example of a maximum of 1024 items.

```
<MaxID>1024</MaxID>
```

5.3.16 MaxMem

Usage: Specifies the maximum memory size for a given local datastore, in bytes.**Parent Elements:** DSMem**Restrictions:** Content information **MUST** be specified as the decimal integer number of maximum free bytes of memory available in the local database. This element type is optional.**Content Model:**

```
(#PCDATA)
```

Attributes: None.**Example:** The following is an example of 65539 bytes.

```
<MaxMem>65539</MaxMem>
```

5.3.17 Mod

Usage: Specifies the model name or model number of the device.**Parent Elements:** DevInf**Restrictions:** This element type is optional.**Content Model:**

```
(#PCDATA)
```

Attributes: None.**Example:**

```
<Mod>1447</Mod>
```

5.3.18 OEM

Usage: Specifies the OEM (Original Equipment Manufacturer) of the device.**Parent Elements:** DevInf**Restrictions:** This element type is optional.**Content Model:**

```
(#PCDATA)
```

Attributes: None.

Example:

```
<OEM>Bar Works, Ltd.</OEM>
```

5.3.19 ParamName

Usage: Specifies supported parameters of a given content type property.

Parent Elements: CTCap

Restrictions: If the content type is either `text/x-vcard`, `text/vcard`, `text/x-vcalendar` or `text/calendar`, the value for this element type MUST be one of the values defined in the table below. Sending the `ParamName` element is optional if the device supports all the parameters of all the supported properties (e.g. property `TEL` is supported, and parameters `PREF`, `WORK`, `HOME`, `VOICE`, `FAX`, `MSG`, `CELL`, `PAGER`, `BBS`, `MODEM`, `CAR`, `ISDN`, and `VIDEO` are all supported).

text/x-vcard	
PropName	ParamName
ADR	DOM, INTL, POSTAL, PARCEL, HOME, WORK
EMAIL	AOL, AppleLink, ATTMail, CIS, eWorld, INTERNET, IBMMail, MCIMail, POWERSHARE, PRODIGY, TLX, X400
LABEL	DOM, INTL, POSTAL, PARCEL, HOME, WORK
TEL	PREF, WORK, HOME, VOICE, FAX, MSG, CELL, PAGER, BBS, MODEM, CAR, ISDN, VIDEO
PHOTO	GIF, CGM, WMF, BMP, MET, PMB, DIB, PICT, TIFF, PS, PDF, JPEG, MPEG, MPEG2, AVI, QTIME
SOUND	WAVE, PCM, AIFF
KEY	X509, PGP
LOGO	GIF, CGM, WMF, BMP, MET, PMB, DIB, PICT, TIFF, PS, PDF, JPEG, MPEG, MPEG2, AVI, QTIME
text/x-vcalendar	
PropName	ParamName
ATTENDEE	ROLE, STATUS, RSVP, EXPECT
AALARM	WAVE, PCM, AIFF
text/vcard	

PropName	ParamName
LOGO	GIF, CGM, WMF, BMP, MET, PMB, DIB, PICT, TIFF, PS, PDF, JPEG, MPEG, MPEG2, AVI, QTIME, other IANA registered image types
LABEL	DOM, INTL, POSTAL, PARCEL, HOME, WORK, other IANA registered parameter names
PHOTO	GIF, CGM, WMF, BMP, MET, PMB, DIB, PICT, TIFF, PS, PDF, JPEG, MPEG, MPEG2, AVI, QTIME, other IANA registered image types
ADR	DOM, INTL, POSTAL, PARCEL, HOME, WORK, other IANA registered parameter names
TEL	PREF, WORK, HOME, VOICE, FAX, MSG, CELL, PAGER, BBS, MODEM, CAR, ISDN, VIDEO, other IANA registered parameter names
EMAIL	PREF, INTERNET, X400, other IANA registered address types
SOUND	WAVE, PCM, AIFF, other IANA registered audio formats
KEY	X509, PGP, other IANA registered certificate types
text/calendar	
PropName	ValEnum
ATTENDEE	CN , CUTYPE , DELEGATED-FROM , DELEGATED-TO , DIR , LANGUAGE , MEMBER , PARTSTAT , ROLE , RSVP , SENT-BY
ORGANIZER	CN , DIR , LANGUAGE , SENT-BY

Content Model:

(#PCDATA)

Attributes: None.

Example: The following is an example of supporting both the HOME and WORK parameters of the vCard TEL property.

```

<CTCap>
  <CTType>text/x-vcard</CTType>
  <PropName>TEL</PropName>
    <ParamName>HOME</ParamName>
    <ParamName>WORK</ParamName>
</CTCap>
    
```

5.3.20 PropName

Usage: Specifies a supported property of a given content type.

Parent Elements: CTCap

Restrictions: If the content type is either `text/x-vcard`, `text/vcard`, `text/x-vcalendar` or `text/calendar`, the value for this element type **MUST** be one of the values defined in the table below.

Content type	PropName
<code>text/x-vcard</code>	BEGIN, VERSION, END, FN, N, PHOTO, BDAY, ADR, LABEL, TEL, EMAIL, MAILER, TZ, GEO, TITLE, ROLE, LOGO, AGENT, ORG, NOTE, REV, SOUND, URL, UID, KEY
<code>text/x-vcalendar</code>	BEGIN, VERSION, END, DAYLIGHT, GEO, PROPID, TZ, ATTACH, ATTENDEE, AALARM, CATEGORIES, CLASS, DCREATED, COMPLETED, DESCRIPTION, DALARM, DUE, DTEND, EXDATE, EXRULE, LAST-MODIFIED, LOCATION, MALARM, RNUM, PRIORITY, PALARM, RELATED-TO, RDATE, RRULE, RESOURCES, SEQUENCE, DTSTART, STATUS, SUMMARY, TRANSP, URL, UID, VALUE, RSVP, ENCODING
<code>text/vcard</code>	BEGIN, VERSION, END, FN, N, NICKNAME, PHOTO, BDAY, ADR, LABEL, TEL, EMAIL, MAILER, TZ, GEO, TITLE, ROLE, LOGO, AGENT, ORG, CATEGORIES, NOTE, REV, SOUND, URL, UID, CLASS, KEY
<code>text/calendar</code>	ALTREP, CN, CUTYPE, DELEGATED-TO, DELEGATED-FROM, DIR, ENCODING, FBTYPE, LANGUAGE, MEMBER, PARTSTAT, RANGE, RELATED, RELTYPE, ROLE, RSVP, TZID, VALUE, BEGIN, END, VERSION, CALSCALE, GEO, METHOD, PROPID, TZ, VERSION, ATTACH, CATEGORIES, CLASS, COMMENT, DESCRIPTION, LOCATION, PERCENT-COMPLETE, PRIORITY, RESOURCES, STATUS, SUMMARY, COMPLETED, DTEND, DUE, DTSTART, DURATION, FREEBUSY, TRANSP, TZNAME, TZOFFSETFROM, TZOFFSETTO, TZURL, ATTENDEE, CONTACT, ORGANIZER, RECURRENCE-ID, RELATED-TO, URL, UID, EXDATE, EXRULE, RDATE, RNUM, RRULE, ACTION, REPEAT, TRIGGER, CREATED, DTSTAMP, LAST-MODIFIED, SEQUENCE, XTOKEN, REQUEST-STATUS

Content Model:

(#PCDATA)

Attributes: None.

Example: The following is an example of supporting properties BEGIN, VERSION, DTSTART, DTEND, DESCRIPTION, END of the `text/x-vcalendar` content type.

```
<CTType>text/x-vcalendar</CTType>
  <PropName>BEGIN</PropName>
    <ValEnum>VCALENDAR</ValEnum>
    <ValEnum>VEVENT</ValEnum>
```

```
<PropName>VERSION</PropName>
  <ValEnum>1.0</ValEnum>
<PropName>DTSTART</PropName>
<PropName>DTEND</PropName>
<PropName>DESCRIPTION</PropName>
<PropName>END</PropName>
  <ValEnum>VCALENDAR</ValEnum>
  <ValEnum>VEVENT</ValEnum>
</CTCap>
```

5.3.21 Rx

Usage: Specifies the supported type and version of a content type received by the device.

Parent Elements: DataStore

Restrictions: This element type is optional.

Content Model:

```
(CTType, VerCT)
```

Attributes: None.

Example:

```
<Rx>
  <CTType>text/x-vcard</CTType>
  <VerCT>2.1</VerCT>
</Rx>
```

5.3.22 Rx-Pref

Usage: Specifies the preferred type and version of a content type received by the device.

Parent Elements: DataStore

Restrictions: The Rx-Pref element type is required for each specified datastore.

Content Model:

```
(CTType, VerCT)
```

Attributes: None.

Example:

```
<Rx-Pref>
```

```
<CTType>text/vcard</CTType>
<VerCT>2.1</VerCT>
</Rx-Pref>
```

5.3.23 SharedMem

Usage: Specifies if the datastore memory is shared. If the memory is shared, the actual memory space is used also by other datastores, and the actual memory space may be more limited than in theory it might be.

Parent Elements: DSMem

Restrictions: The content of this element MUST be empty. This element type is used as a flag, and if this element type is present, then the given datastore memory is shared. This element is optional.

Content Model:

```
EMPTY
```

Attributes: None.

Example: The following is an example of shared datastore memory.

```
<DSMem>
  <SharedMem/>
  <MaxMem>65539</MaxMem>
  <MaxID>512</MaxID>
</DSMem>
```

5.3.24 Size

Usage: Specifies the size of a property or a parameter of a given content type.

Parent Elements: CTCap

Restrictions: If the value of the DataType element is either int or bin, then the Size element type is used to specify the size of the property or parameter.

Content Model:

```
(#PCDATA)
```

Attributes: None.

Example: The following is an example of supporting the vCard property PHOTO, and the size of the GIF image is 2400 bytes.

```
<PropName>PHOTO</PropName>
  <ParamName>GIF</PropName>
  <DataType>bin</DataType>
  <Size>2400</Size>
```


5.3.25 SourceRef

Usage: Specifies the reference URI for a local database.

Parent Elements: DataStore

Restrictions: If the DataStore element type is present, then the SourceRef element type is required. The content information of this element type is either the absolute or relative URI of the datastore.

Content Model:

```
(#PCDATA)
```

Attributes: None.

Example: The following is an example of a source reference to a relative URI for the InBox database.

```
<SourceRef>./InBox</SourceRef>
```

5.3.26 SupportLargeObjs

Usage: Specifies that the device supports handling of large objects.

Parent Element: DevInf

Restrictions: If the sending device has specified <SupportLargeObjs/> elements in its Device Information then the receiving device MUST (for server) or SHOULD (for client) specify <MaxObjSize> in its Meta Information as specified in the Meta Information specification v1.1 [SYNCMETA].

Content Model:

```
(EMPTY)
```

Attributes: None.

Example:

```
<DevInf>
  <VerDTD>1.1</VerDTD>
  <DevID>1218182THD012345-2</DevID>
  <DevTyp>pager</DevTyp>
  <SupportLargeObjs/>
  <DataStore>
    ...
  </DataStore>
</DevInf>
```

5.3.27 SupportNumberOfChanges

Usage: Specifies that the device supports number of changes

ParentElement: DevInf

Restrictions: Server SHOULD send <NumberOfChanges>, specified in Representation protocol specification v1.1 [DSREPU], if the client specifies <SupportNumberOfChanges/>. Server MUST NOT send <NumberOfChanges> if the client has not specified <SupportNumberOfChanges/> element in its Device Information.

Content Model:

(EMPTY)

Attributes: None.

Example:

```
(<DevInf>
  <VerDTD>1.1</VerDTD>
  <DevID>1218182THD012345-2</DevID>
  <DevTyp>pager</DevTyp>
  <SupportNumberOfChanges/>
  <DataStore>
    ...
  </DataStore>
</DevInf>
```

5.3.28 SwV

Usage: Specifies the software version of the device.

Parent Elements: DevInf

Restrictions: If there is no software version of the device available, then the content information can also be a date. If the content information is a date, then it MUST be formatted as a complete representation, basic format of a [DEIF] date or date and UTC time of day. For example, 19980114 or 19990714T133000Z. Only hours, minutes and second MUST be specified in the time component. This element type is optional.

Content Model:

(#PCDATA)

Attributes: None.

Example:

```
<SwV>0.1a</SwV>
```

5.3.29 SyncCap

Usage: Specifies the synchronization capabilities of the given local datastore.

Parent Elements: DataStore

Restrictions: This element is mandatory.

Content Model:

(SyncType+)

Attributes: None.

Example: The following is an example of supporting both two-way, and server alerted sync.

```
<DataStore>
...
  <SyncCap>
    <SyncType>1</SyncType>
    <SyncType>7</SyncType>
  </SyncCap>
</DataStore>
```

5.3.30 SyncType

Usage: Specifies the type of the supported synchronization.

Parent Elements: SyncCap

Restrictions: If the device supports synchronizations specified in the SyncML Sync protocol specification [SYNCPRO], then the value of this element MUST be one of the synchronization types specified in the table below. Other values can also be specified. One or more of the element types are required. One element type is required for each of the supported synchronizations.

Support of 'two-way sync'	'1'
Support of 'slow two-way sync'	'2'
Support of 'one-way sync from client only'	'3'
Support of 'refresh sync from client only'	'4'
Support of 'one-way sync from server only'	'5'
Support of 'refresh sync from server only'	'6'
Support of 'server alerted sync'	'7'

Content Model:

(#PCDATA)

Attributes: None.

Example: The following is an example of identifying support for two-way sync.

```
<SyncType>1</SyncType>
```

5.3.31 Tx

Usage: Specifies the supported type and version of a content type transmitted by the device.

Parent Elements: DataStore

Restrictions: This element type is optional.

Content Model:

(CTType, VerCT)

Attributes: None.

Example:

```
<Tx>
  <CTType>text/x-vcard</CTType>
  <VerCT>2.1</VerCT>
</Tx>
```

5.3.32 Tx-Pref

Usage: Specifies the preferred type and version of a content type transmitted by the device.

Parent Elements: DataStore

Restrictions: The Tx-Pref element type is required for each specified datastore.

Content Model:

(CTType, VerCT)

Attributes: None.

Example:

```
<Tx-Pref>
  <CTType>text/vcard</CTType>
  <VerCT>3.0</VerCT>
</Tx-Pref>
```

5.3.33 UTC

Usage: Specifies that the device supports UTC based time.

Parent Element: DevInf

Restrictions: If UTC flag is present, the server SHOULD send time in UTC form, else MUST send in local time. Client MAY send time in local or UTC format.

Content Model:

(EMPTY)

Attributes: None.

Example:

```

<DevInf>
  <VerDTD>1.1</VerDTD>
  <DevID>1218182THD012345-2</DevID>
  <DevTyp>pager</DevTyp>
  <UTC/>
  <DataStore>
    ...
  </DataStore>
</DevInf>
    
```

5.3.34 ValEnum

Usage: Specifies the supported enumerated value of a given content type property.

Parent Elements: CTCap

Restrictions: If the content type is either `text/x-vcard`, `text/vcard`, `text/x-vcalendar` or `text/calendar`, the value for this element type **MUST** be one of the values defined in the table below.

text/x-vcard	
PropName	ValEnum
BEGIN	VCARD
END	VCARD
VERSION	2.1
text/x-vcalendar	
PropName	ValEnum
BEGIN	VCALENDAR, VEVENT, VTODO
END	VCALENDAR, VEVENT, VTODO
VERSION	1.0
CLASS	PUBLIC, PRIVATE, CONFIDENTIAL
text/vcard	
PropName	ValEnum

BEGIN	VCARD
END	VCARD
VERSION	3.0
CLASS	PUBLIC, PRIVATE, CONFIDENTIAL
text/calendar	
PropName	ValEnum
CUTYPE	INDIVIDUAL, GROUP, RESOURCE, ROOM, UNKNOWN
BEGIN	VCALENDAR, VEVENT, VTODO, VJOURNAL, VFREEBUSY, VTIMEZONE, VALARM
END	VCALENDAR, VEVENT, VTODO, VJOURNAL, VFREEBUSY, VTIMEZONE, VALARM
VERSION	2.0
ATTACH	URI, BINARY
CLASS	PUBLIC, PRIVATE, CONFIDENTIAL
ACTION	AUDIO, DISPLAY, EMAIL, PROCEDURE
RELTYPE	PARENT, CHILD, SIBLING

Content Model:

(#PCDATA)

Attributes: None.**Example:** The following is an example of supporting iCalendar binary attachments.

```

<CTCap>
  <CTType>text/calendar</CTType>
  <PropName>BEGIN</PropName>
    <ValEnum>VCALENDAR</ValEnum>
    <ValEnum>VEVENT</ValEnum>
  <PropName>VERSION</PropName>
    <ValEnum>2.0</ValEnum>
  <PropName>ATTACH<PropName>
    <ValEnum>BINARY</ValEnum>
</CTCap>

```

5.3.35 VerCT

Usage: Specifies the version of a supported content type.

Parent Elements: Rx, Rx-Pref, Tx, Tx-Pref

Restrictions: If a parent element is present, this element type is required.

Content Model:

```
(#PCDATA)
```

Attributes: None.

Example:

```
<Rx>
  <CTType>text/x-vcard</CTType>
  <VerCT>2.1</VerCT>
</Rx>
```

5.3.36 VerDTD

Usage: Specifies the major and minor version identifier of the Device Information DTD used in the representation of the Device Information document.

Parent Elements: DevInf

Restrictions: Major revisions of the specification create incompatible changes that will generally require a new parser. Minor revisions involve changes that do not impact basic compatibility of the parser. When the XML document conforms to this revision of the Device Information specification the value MUST be 1.1. The element type MUST be included in the DevInf element.

Content Model:

```
(#PCDATA)
```

Attributes: None.

Example:

```
<DevInf>
  <VerDTD>1.1</VerDTD>
  ...
</DevInf>
```

5.3.37 XNam

Usage: Specifies the name of one of the DevInf extension element types.

Parent Elements: Ext

Restrictions: The element type is required whenever an Ext element is present.

Content Model:

```
(#PCDATA)
```

Attributes: None.

Example:

```
<Ext>
  <XNam>X-Foo-Cliver</XNam>
  <XVal>5.0</Xval>
  <XVal>5.01</Xval>
  <XVal>5.02<Xval>
</Ext>
```

5.3.38 XVal

Usage: Specifies one of the valid values for a DevInf extension element type.

Parent Elements: Ext

Restrictions: One or more of the element types are required whenever an Ext element is present. One element type is required for each of the valid values for the extension element type. Ranges of valid values can be specified by a sequence of the first value in the range, followed by the string ".." (i.e., PERIOD PERIOD), followed by the last value in the range.

Content Model:

```
(#PCDATA)
```

Attributes: None.

Example: The following example is for an extension element type that has a range of valid integer values from 1 to 5.

```
<Ext>
  <XNam>X-Bar-Enum</XNam>
  <XVal>1</XVal>
  <XVal>..</XVal>
  <XVal>5<XVal>
</Ext>
```


6. Device Information DTD

```

<!--
SyncML Device Information (SYNCML-DEVINF) V1.1 Document Type Definition

Copyright Open Mobile Alliance Ltd., 2002-2003
All rights reserved

This DTD defines device information that is used within
the SyncML Representation Protocol DTD. Typical usage:
  <!DOCTYPE DevInf PUBLIC "-//OMA//DTD SYNCML-DEVINF 1.1//EN"
    "http://www.openmobilealliance.org/DTD/OMA-SyncML-
Device_Information-DTD-1.1.2-20030221-D.dtd"
    [<?oma-syncml-devinf-ver supported-versions="1.1"?>]>
  <DevInf>
    ...
  </DevInf>

Terms and conditions of use are available from the
Open Mobile Alliance Ltd. web site at
http://www.openmobilealliance.org/useterms.html
-->

<!-- Root element -->
<!ELEMENT DevInf (VerDTD, Man?, Mod?, OEM?, FwV?, SwV?, HwV?, DevID,
DevTyp, DataStore+, CTCap*, Ext*)>
<!ELEMENT Man (#PCDATA)>
<!ELEMENT Mod (#PCDATA)>
<!ELEMENT OEM (#PCDATA)>
<!ELEMENT FwV (#PCDATA)>
<!ELEMENT SwV (#PCDATA)>
<!ELEMENT HwV (#PCDATA)>
<!ELEMENT DevID (#PCDATA)>
<!ELEMENT DevTyp (#PCDATA)>
<!ELEMENT DataStore (SourceRef, DisplayName?, MaxGUIDSize?, Rx-Pref, Rx*,
Tx-Pref, Tx*, DSMem?, SyncCap)>
<!ELEMENT MaxGUIDSize (#PCDATA)>
<!ELEMENT Rx-Pref (CTType, VerCT)>
<!ELEMENT Rx (CTType, VerCT)>
<!ELEMENT Tx-Pref (CTType, VerCT)>
<!ELEMENT Tx (CTType, VerCT)>
<!ELEMENT DSMem (SharedMem?, MaxMem?, MaxID?)>
<!ELEMENT SharedMem EMPTY>
<!ELEMENT SourceRef (#PCDATA)>
<!ELEMENT DisplayName (#PCDATA)>
<!ELEMENT CTCap ((CTType, (PropName, (ValEnum+ | (DataType,
Size?))?, DisplayName?, (ParamName, (ValEnum+ | (DataType,
Size?))?, DisplayName?)*)+)>
<!ELEMENT CTType (#PCDATA)>
<!ELEMENT DataType (#PCDATA)>
<!ELEMENT Size (#PCDATA)>
<!ELEMENT PropName (#PCDATA)>
<!ELEMENT ValEnum (#PCDATA)>
<!ELEMENT ParamName (#PCDATA)>
<!ELEMENT SyncCap (SyncType+)>
<!ELEMENT SyncType (#PCDATA)>
<!ELEMENT Ext (XNam, XVal*)>

```

```
<!ELEMENT XNam (#PCDATA)>
<!ELEMENT XVal (#PCDATA)>
<!ELEMENT MaxMem (#PCDATA)>
<!ELEMENT MaxID (#PCDATA)>
<!ELEMENT VerCT (#PCDATA)>
<!ELEMENT VerDTD (#PCDATA)>
<!-- New Elements are added here -->
<!ELEMENT UTC EMPTY>
<!ELEMENT SupportLargeObjs EMPTY>
<!ELEMENT SupportNumberOfChanges EMPTY>
<!-- End of DTD -->
```

7. WBXML Definitions

This version of the DevInf DTD specification is associated with the WBXML code space FD4 and the formal public identifier -//SYNCML//DTD DevInf 1.1//EN.

The following WBXML token codes represent element types (i.e., tags) form code page 0x00, DevInf DTD.

Element Type Name	WBXML Tag Token (Hex Value)
CTCap	05
CTType	06
DataStore	07
DataType	08
DevID	09
DevInf	0A
DevTyp	0B
DisplayName	0C
DSMem	0D
Ext	0E
FwV	0F
HwV	10
Man	11
MaxGUIDSize	12
MaxID	13
MaxMem	14
Mod	15
OEM	16
ParamName	17
PropName	18
Rx	19
Rx-Pref	1A
SharedMem	1B
Size	1C

SourceRef	1D
SwV	1E
SyncCap	1F
SyncType	20
Tx	21
Tx-Pref	22
ValEnum	23
VerCT	24
VerDTD	25
Xnam	26
Xval	27
UTC	28
SupportNumberOfChanges	29
SupportLargeObjs	2A

8. EXAMPLES

The following is an example of a XML representation for the device information object. A XML representation of a device information object that conforms to this specification must include the name space definition on the `devinf` element type.

```
<DevInf xmlns='syncml:devinf'>
  <VerDTD>1.1</VerDTD>
  <Man>Big Factory, Ltd.</Man>
  <Mod>4119</Mod>
  <OEM>Jane's phones</OEM>
  <FwV>2.0e</FwV>
  <SwV>2.0</SwV>
  <HwV>1.22I</HwV>
  <DevID>1218182THD000001-2</DevID>
  <DevTyp>phone</DevTyp>
  <UTC/>

  <SupportLargeObjs/>

  <SupportNumberOfChanges/>

  <DataStore>
    <SourceRef>./contacts</SourceRef>
    <DisplayName>Phonebook</DisplayName>
    <MaxGUIDSize>32</MaxGUIDSize>
    <Rx-Pref>
      <CTType>text/vcard</CTType>
      <VerCT>3.0</VerCT>
    </Rx-Pref>
    <Tx-Pref>
      <CTType>text/vcard</CTType>
      <VerCT>3.0</VerCT>
    </Tx-Pref>
    <Tx>
      <CTType>text/x-vcard</CTType>
      <VerCT>2.1</VerCT>
    </Tx>
    <DSMem>
      <MaxMem>32650</MaxMem>
      <MaxID>250</MaxID>
  </DataStore>
</DevInf>
```

```
</DSMem>
<SyncCap>
  <SyncType>1</SyncType>
  <SyncType>7</SyncType>
</SyncCap>
</DataStore>
<CTCap>
  <CTType>text/x-vcard</CTType>
  <PropName>BEGIN</PropName>
    <ValEnum>VCARD</ValEnum>
  <PropName>END</PropName>
    <ValEnum>VCARD</ValEnum>
  <PropName>VERSION</PropName>
    <ValEnum>2.1</ValEnum>
  <PropName>N</PropName>
  <PropName>TEL</PropName>
    <ParamName>VOICE</ParamName>
    <ParamName>FAX</ParamName>
    <ParamName>CELL</ParamName>
  <CTType>text/vcard</CTType>
  <PropName>BEGIN</PropName>
    <ValEnum>VCARD</ValEnum>
  <PropName>END</PropName>
    <ValEnum>VCARD</ValEnum>
  <PropName>VERSION</PropName>
    <ValEnum>3.0</ValEnum>
  <PropName>N</PropName>
  <PropName>TEL</PropName>
    <ParamName>VOICE</ParamName>
    <ParamName>FAX</ParamName>
    <ParamName>CELL</ParamName>
</CTCap>
```

```
<Ext>
  <XNam>srtmsg</XNam>
  <XVal>Hello World</XVal>
</Ext>
<Ext>
  <XNam>endmsg</XNam>
  <XVal>Goodbye</XVal>
</Ext>
</DevInf>
```

9. MIME Media Type Registration

The following section is the MIME media type registrations for SyncML Device Information specific MIME media types.

9.1 application/vnd.syncml-devinf+xml

To: ietf-types@iana.org

Subject: Registration of MIME media type application/vnd.syncml-devinf+xml

MIME media type name: application

MIME subtype name: vnd.syncml-devinf+xml

Required parameters: None

Optional parameters: charset, verDTD

Content-Type MIME header.

charset Parameter

Purpose: Specifies the character set used to represent the Device Information document. The default character set for SyncML Device Information document is UTF-8, as defined [RFC 2279].

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

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

Interoperability considerations: Implementations that have support for the mandatory features of this content type will greatly increase the chances of interoperating with other implementations supporting this content type. Conformance to this content type requires an implementation to support every mandatory feature.

verDTD Parameter

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

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

```
verDTD-param = ";" "verDTD" "=" 1*numeric "." 1*numeric
```

```
text = 1*ALPHA
```

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


Published specification:
http://www.syncml.org/docs/syncml_devinf_v11_20020215.pdf

Applications, which use this media type: This MIME content type is intended for common use by networked data synchronization applications.

Additional information:

Magic number(s): None

File extension(s): XDM

Macintosh File Type Code(s): XDML

Person & email address to contact for further information:
admins@syncml.org

Intended usage: COMMON

Author/Change controller: admins@syncml.org

9.2 application/vnd.syncml-devinf+wbxml

To: ietf-types@iana.org

Subject: Registration of MIME media type application/vnd.syncml-devinf+wbxml

MIME media type name: application

MIME subtype name: vnd.syncml-devinf+wbxml

Required parameters: None

Optional parameters: charset, verDTD

Content-Type MIME header.

charset Parameter

Purpose: Specifies the character set used to represent the Device Information document. The default character set for SyncML Device Information document is UTF-8, as defined [RFC 2279].

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

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

Interoperability considerations: Implementations that have support for the mandatory features of this content type will greatly increase the chances of interoperating with other implementations supporting this content type. Conformance to this content type requires an implementation to support every mandatory feature.

verDTD Parameter

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

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

```
vertdtd-param = ";" "vertdtd" "=" 1*numeric "." 1*numeric
```

```
text = 1*ALPHA
```

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

Published specification:

http://www.syncml.org/docs/syncml_devinf_v11_20020215.pdf

Applications, which use this media type: This MIME content type is intended for common use by networked data synchronization applications.

Additional information:

Magic number(s): None

File extension(s): BDM

Macintosh File Type Code(s): BDML

Person & email address to contact for further information:

admins@syncml.org

Intended usage: COMMON

Author/Change controller: admins@syncml.org

Appendix A. Static Conformance Requirements (Normative)

Static conformance requirements (SCR) specify the features that are optional, mandatory and recommended within implementations conforming to this specification.

Simple tables are used to specify this information

In these tables, optional features are specified by a "MAY", mandatory features are specified by a "MUST" and recommended features are specified by a "SHOULD", as defined in [[RFC2199]].

An implementation, which does not include a particular option, MUST be prepared to interoperate with another implementation, which does include the option, though perhaps with reduced functionality.

The following specifies the static conformance requirements for the SyncML Device Information element types for devices conforming to this specification.

Element Type	Support of Synchronization Server		Support of Synchronization Client	
	Sending	Receiving	Sending	Receiving
CTCap	SHOULD	MUST	SHOULD	SHOULD
CTType	MUST	MUST	MUST	MUST
DataStore	MUST	MUST	MUST	MUST
DataType	MAY	MUST	MAY	MAY
DevID	MUST	MUST	MUST	MUST
DevInf	MUST	MUST	MUST	MUST
DevTyp	MUST	MUST	MUST	MUST
DisplayName	MAY	MAY	MAY	MAY
DSMem	MAY	SHOULD	SHOULD	MAY
Ext	MAY	MAY	MAY	MAY
FwV	MAY	SHOULD	SHOULD	MAY
HwV	MAY	SHOULD	SHOULD	MAY
Man	MAY	SHOULD	SHOULD	MAY
MaxGUIDSize	MUST NOT	MUST	MUST	MUST NOT
MaxID	MAY	SHOULD	SHOULD	MAY
MaxMem	MAY	SHOULD	SHOULD	MAY
Mod	MAY	MAY	MAY	MAY
OEM	MAY	MAY	MAY	MAY

ParamName	SHOULD	MUST	SHOULD	SHOULD
PropName	SHOULD	MUST	MUST	SHOULD
Rx	MAY	MUST	MAY	MUST
Rx-Pref	MUST	MUST	MUST	MUST
SharedMem	SHOULD	MAY	SHOULD	MAY
Size	MAY	MUST	MAY	MAY
SourceRef	MUST	MUST	MUST	MUST
SupportLargeObjs	MUST	MUST	SHOULD	SHOULD
SupportNumberOfChanges	MAY	MUST	MAY	MAY
SwV	MAY	SHOULD	SHOULD	MAY
SyncCap	MUST	MUST	MUST	MUST
SyncType	MUST	MUST	MUST	MUST
Tx	MAY	MUST	MAY	MUST
Tx-Pref	MUST	MUST	MUST	MUST
UTC	MAY	MUST	MAY	MAY
ValEnum	SHOULD	MUST	MUST	SHOULD
VerCT	MUST	MUST	MUST	MUST
VerDTD	MUST	MUST	MUST	MUST
Xnam	MAY	MAY	MAY	MAY
Xval	MAY	MAY	MAY	MAY

Appendix B. Change History

(Informative)

B.1 Approved Version History

Reference	Date	Description
OMA-SyncML-DevInfo_V1_1_2-20030612-A	12 June 2003	Approved by TP. TP ref# OMA-TP-2003-0264R1