OMA DS Device Information

Approved Version 1.2 – 10 Jul 2006

Open Mobile Alliance

OMA-TS-DS_DevInf-V1_2-20060710-A

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.

Contents

Figures

Tables

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 and device management. 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 and device management protocols has placed barriers to the extended use of mobile devices, has restricted data access and delivery and limited the mobility of the users.

The SyncML Initiative merged with the Open Mobile Alliance in November 2002. The SyncML legacy specifications were converted to the OMA format with the 1.1.2 versions of OMA SyncML Common, OMA Data Synchronization and OMA Device Management in May 2002. The relationship between these documents which had been created during the SyncML Initiative has been preserved and is depicted in Figure 1 OMA SyncML Specification Structure and Relationships.

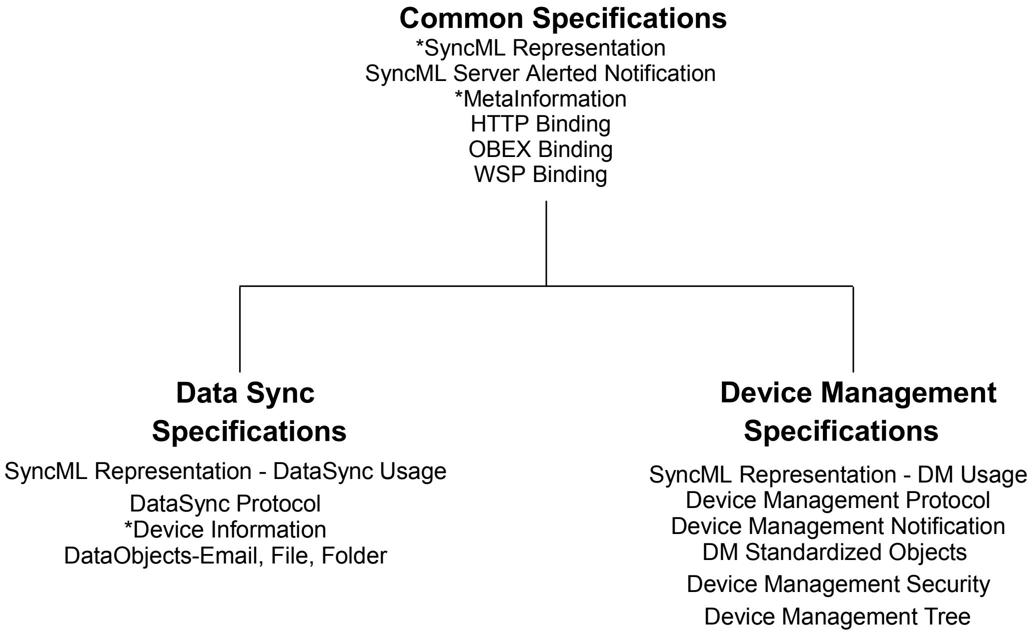
Common Specifications
*SyncML Representation
SyncML Server Alerted Notification
*MetaInformation
HTTP Binding
OBEX Binding
WSP Binding
Data Sync Specifications
SyncML Representation - DataSync Usage
DataSync Protocol
*Device Information
DataObjects-Email, File, Folder
Device Management Specifications
SyncML Representation - DM Usage
Device Management Protocol
Device Management Notification
DM Standardized Objects
Device Management Security
Device Management Tree

Figure 1 OMA SyncML Specification Structure and Relationships

The OMA SyncML Common Specifications Enabler Release includes the following documents:

- SyncML Representation: The XML-based representation protocol which specifies the common XML syntax and semantics used by all SyncML protocols. This defines the superset of the DS and DM representation protocols. (* includes DTD).
- The transport bindings: HTTP, OBEX, WSP. These specify the features REQUIRED for each transport to send and receive DS and DM protocol messages.
- The Meta Information associated with a SyncML command or data item or collection used by either DS or DM (* includes DTD)
- SyncML Server Alerted Notification: The logical structure and format of the notification messages used by all SyncML server alerted notifications, for both DS and DM.

The OMA Data Synchronization Specifications Enabler Release includes the following documents:

- SyncML Representation DataSync Usage: The subset of the Common Specifications SyncML Representation Specification necessary to define the Data Synchronization commands and protocol, with examples and commentary specific to DS.
- DataSyncProtocol: Specifies how SyncML Common messages conforming to the DTD are exchanged in order to allow an OMA DS client and server to exchange additions, deletions, updates and other status information.
- Device Information: Used to exchange device specific information, including hardware, firmware, software levels, available memory, and local databases supported. (* Includes DTD)
- Data Objects: Email, File, Folder: Each object is identified by a unique MIME media type (eg. **application/vnd.omads-email)**. The objects are either represented by or encapsulated in a mark-up language defined by xml. Meta or state data is included in the representation (eg. Read/Unread, Creation Date, Last Modified Date).

Although the SyncML Common specification defines transport bindings that specify how to use a particular transport to exchange messages and responses, the SyncML Common representation, synchronization and device management protocols are transport-independent. Each package in these protocols 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 Common could not be implemented using email or message queues, to list only two alternatives. Because the SyncML Common messages are self-contained, multiple transports could 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.

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

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

This document specifies the device information syntax and semantics used by the OMA 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

[DSREPU] "SyncML Representation Protocol, Data Synchronization Usage, Open Mobile Alliance™, OMA-TS-DS_DataSyncRep-V1_2", URL:http:www.openmobilealliance.org.

[META] "SyncML Meta Information", Open Mobile Alliance™, OMA-SyncML-MetaInformation-V1_2", URL:http:www.openmobilealliance.org.

[RFC2045] "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies" N. Freed & N. Borenstein. URL: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

[DSPRO] "DS Protocol", Open Mobile Alliance™, OMATS-DS_Protocol-V1_2", URL:http:www.openmobilealliance.org.

[WBXML] "WAP Binary XML Content Format Specification", WAP Forum™, WAP-154-WBXML, URL:http://www.openmobilealliance.org

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 An OMA DS application that supports the OMA DS protocol. The application can either be the originator or recipient of the SyncML protocol commands. The application can act as an OMA DS client or an OMA DS server.

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

Client An OMA DS 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, an OMA DS 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 An OMA DS 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 [**Error! Reference source not found.**]. 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 1.2//EN
```

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

```
devinf12
```

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 [**Error! Reference source not found.**]. The token values used to encode the Device Information documents are defined in chapter 7 of this document. The use of [**Error! Reference source not found.**] 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: `DataStore`

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

Content Model:

```
CTCap ( CTType, VerCT, FieldLevel?, Property+)
```

Attributes: None.

Example:

```
<CTCap>
  <CTType>text/x-vcard</CTType>
  <VerCT>2.1</VerCT>
<Property>
    <PropName>BEGIN</PropName>
    <ValEnum>VCARD</ValEnum>
  </Property>
  <Property>
    <PropName>END</PropName>
    <ValEnum>VCARD</ValEnum>
  </Property>
  <Property>
    <PropName>VERSION</PropName>
    <ValEnum>2.1</ValEnum>
  </Property>
  <Property>
```

```
    <PropName>N</PropName>
  </Property>
  <Property>
    <PropName>TEL</PropName>
    <PropParam>
      <ParamName>TYPE</ParamName>
      <ValEnum>VOICE,HOME</ValEnum>
      <ValEnum>FAX,HOME</ValEnum>
      <ValEnum>VOICE,CELL</ValEnum>
    </PropParam>
  </Property>
</CTCap>
```

5.3.2 CTType

Usage: Specifies the type of a supported content type.

Parent Elements: `CTCap, Filter-Rx, FilterCap, Rx, Rx-Pref, Tx, Tx-Pref`

Restrictions: If a parent element is present, this element type is required. Possible values for this element are specified in the section “Base Media and Content Formats” of SyncML representation protocol, Data Synchronization Usage [DSREPU]. Other values can also be specified.

Content Model:

```
CTType (#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:

```
DataStore (SourceRef, DisplayName?, MaxGUIDSize?, Rx-Pref, Rx*, Tx-Pref,
Tx*, CTCap+, DSMem?, SupportHierarchicalSync?, SyncCap, Filter-Rx*,
FilterCap*)
```

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/x-vcard</CTType>
    <VerCT>3.0</VerCT>
  </Tx-Pref>
  <Tx>
    <CTType>text/x-vcard</CTType>
    <VerCT>2.1</VerCT>
  </Tx>
  <CTCap>
    <CTType>text/vcard</CTType>
    <VerCT>3.0</VerCT>
    <Property>
      <PropName>BEGIN</PropName>
      <ValEnum>VCARD</ValEnum>
    </Property>
    <Property>
      <PropName>END</PropName>
      <ValEnum>VCARD</ValEnum>
    </Property>
    <Property>
      <PropName>VERSION</PropName>
      <ValEnum>3.0</ValEnum>
    </Property>
    <Property>
      <PropName>N</PropName>
    </Property>
    <Property>
      <PropName>TEL</PropName>
      <PropParam>
        <ParamName>TYPE</ParamName>
        <ValEnum>VOICE,HOME</ValEnum>
        <ValEnum>FAX, HOME</ValEnum>
        <ValEnum>VOICE, CELL</ValEnum>
      </PropParam>
    </Property>
  </CTCap>
  <CTCap>
    <CTType>text/vcard</CTType>
    <VerCT>2.1</VerCT>
    <Property>
      <PropName>BEGIN</PropName>
      <ValEnum>VCARD</ValEnum>
    </Property>
    <Property>
      <PropName>END</PropName>
      <ValEnum>VCARD</ValEnum>

    </Property>

    <Property>

      <PropName>VERSION</PropName>

      <ValEnum>2.1</ValEnum>

    </Property>

    <Property>

      <PropName>N</PropName>

    </Property>

    <Property>

      <PropName>TEL</PropName>

      <PropParam>

        <ParamName>TYPE</ParamName>

        <ValEnum>VOICE,HOME</ValEnum>

        <ValEnum>FAX,HOME</ValEnum>

        <ValEnum>VOICE,CELL</ValEnum>

      </PropParam>

    </Property>

  </CTCap>

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

  <SyncCap>

    <SyncType>1</SyncType>

    <SyncType>2</SyncType>

    <SyncType>7</SyncType>

  </SyncCap>

</DataStore>
```

5.3.4 DataType

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

Parent Elements: `Property, PropParam`

Restrictions: Type values for this element type are specified together with the relevant definition of the content type (e.g. for email). Datatype MUST NOT be used for Versit types.

Content Model:

```
DataType (#PCDATA)
```

Attributes: None.

Example:

```
<Property>
  <PropName>read</PropName>
  <DataType>bool</DataType>
</Property>
```

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. For servers the value of this identifier MUST be the same value as used for ServerID used for Server Alerted Notification.

Content Model:

```
DevID (#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:

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

Attributes:

Name	Type	Occurrence	Description
xmlns	CDATA	IMPLICIT	Must be present with a value of

			'syncml:devinf'.

Example:

```
<DevInf xmlns=’syncml:devinf’>
   <VerDTD>1.2</VerDTD>
   <Man>SomeManufacturer</Man>
   <Mod>SomeModel</Mod>
   <OEM>MyOEM</OEM>
   <FwV>1.0</FwV>
   <SwV>2.99</SwV>
   <HwV>2.1</HwV>
   <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:

```
DevTyp (#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, Property, PropParam`

Restrictions: This element type is optional.

Content Model:

```
DisplayName (#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:

```
DSMem (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:

```
Ext (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 FieldLevel

Usage: Presence of <FieldLevel/> element in CTCap indicates that the sender is able to apply field-level replaces for the corresponding CTType.

Parent Elements: `CTCap`

Restrictions: If the sending device has not specified the <FieldLevel/> element in its Device Information for a particular CTType, then the receiving device MUST NOT send field-level changes.

Content Model:

```
FieldLevel EMPTY
```

Attributes: None.

Example:

```
<CTCap>
  <CTType>x-type/x-subtype</CTType>
  <VerCT>2.1</VerCT>
  <FieldLevel/>
  <Property>
    <PropName>Field1</PropName>
    <ValEnum>Field1PossibleValue1</ValEnum>
    <ValEnum>Field1PossibleValue2</ValEnum>
  </Property>
  <Property>
    <PropName>Field2</PropName>
  </Property>
  <Property>
    <PropName>Field3</PropName>
    <PropParam>
      <ParamName>Subfield1</ParamName>
      <ValEnum>PossibleSubfieldValue1</ValEnum>
      <ValEnum>PossibleSubfieldValue2</ValEnum>
      <ValEnum>PossibleSubfieldValue3</ValEnum>
    </PropParam>
  </Property>
</CTCap>
```

5.3.12 Filter-Rx

Usage: Specifies the supported filter grammars that can be received by the data store.

Parent Elements: `Datastore`

Restrictions: If a device supports filtering for a specific data store, then at least one `Filter-Rx` element MUST be present and it MUST support at least the "syncml:filtertype-cgi" grammar. The following example shows the minimum requirements for a device that supports filtering on a specific data store.

Content Model:

```
Filter-Rx (CTType, VerCT)
```

Attributes: None.

Example:

```
<Datastore>
  ...
  <Filter-Rx>
    <CTType>syncml:filtertype-cgi</CTType>
    <VerCT>1.0</VerCT>
  </Filter-Rx>
  ...
```

```
</Datastore>
```

5.3.13 FilterCap

Usage: Indicates the filtering capabilities.

Parent Elements: `Datastore`

Restrictions: For every `Filter-Rx` element, there MUST be a `FilterCap` element containing `CTType` and `VerCT` elements matching the `CTType` and `VerCT` elements specified in the `Filter-Rx` element. Adding a `FilterCap` element without any `FilterKeyword` or any `PropName` elements signifies that record level filtering is unsupported but field level filtering is.

Content Model:

```
FilterCap (CTType, VerCT, FilterKeyword*, PropName*)
```

Attributes: None.

Example:

```
<Datastore>
  ...
  <FilterCap>
    <CTType>syncml:filtertype-cgi</CTType>
    <VerCT>1.0</VerCT>
  </FilterCap>
  ...
</Datastore>
```

5.3.14 FilterKeyword

Usage: Indicates a record level filter keyword that can be used in the `Filter Record` queries.

Parent Elements: `FilterCap`

Restrictions: These keywords MUST be either one of the generic filter keywords listed below or a valid property name for any base media types specified in the `Datastore CTCap` elements.

Keywords	Description	Type
BEFORE	Items whose date is earlier than the specified date	Date
SINCE	Items whose date is within or later than the specified date	Date
UNSEEN	Items that are flagged as “unseen”	Boolean

GROUP	Items that are part of the specified group or category. Note: since vCard 2.1 does not support the CATEGORIES property, this keyword can be used to filter contacts based on CATEGORIES in both vCard 2.1 and vCard 3.0	Text

Content Model:

```
FilterKeyword (#PCDATA)
```

Attributes: None.

5.3.15 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.

Content Model:

```
FwV (#PCDATA)
```

Attributes: None.

Example:

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

5.3.16 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.

Content Model:

```
HwV (#PCDATA)
```

Attributes: None.

Example:

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

5.3.17 Man

Usage: Specifies the name of the manufacturer of the device.

Parent Elements: `DevInf`

Restrictions: None.

Content Model:

```
Man (#PCDATA)
```

Attributes: None.

Example:

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

5.3.18 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:

```
MaxGUIDSize (#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>
```

5.3.19 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:

```
MaxID (#PCDATA)
```

Attributes: None.

Example: The following is an example of a maximum of 1024 items.

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

5.3.20 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:

```
MaxMem (#PCDATA)
```

Attributes: None.

Example: The following is an example of 65539 bytes.

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

5.3.21 MaxOccur

Usage: The maximum number of occurrences of a property of the same type supported within a single object.

Parent Elements: `Property`

Restrictions: Optional tag. If the sending device has specified the <MaxOccur> element in its Device Information for a particular field, then the receiving device MUST NOT (for server) or SHOULD NOT (for client) send more than the specified number of property values for this property.

Content Model:

```
MaxOccur (#PCDATA)
```

Attributes: None.

Example:

```
<Property>
  <PropName>TEL</PropName>
  <MaxOccur>3</MaxOccur>
  <PropParam>
    <ParamName>TYPE</ParamName>
    <ValEnum>VOICE,HOME</ValEnum>
    <ValEnum>FAX,HOME</ValEnum>
    <ValEnum>VOICE,CELL</ValEnum>
  </PropParam>

</Property>
```

5.3.22 MaxSize

Usage: Specifies the maximum size in UTF-8 characters of a given property value. The sender MUST truncate the property to the specified size if the NoTruncate flag is not set. If the NoTruncate flag is set and the property value exceeds the maximum size, the property MUST NOT be sent to the client. If the maximum size is 0, this behaviour will also be observed. In that case if the NoTruncate flag is not set, a property with an empty value MUST be sent.

Parent Elements: `Property`

Restrictions: Optional tag

Content Model:

```
MaxSize (#PCDATA)
```

Attributes: None.

Example:

```
<Property>

  <PropName>TEL</PropName>

  <MaxSize>256</MaxSize>

  <PropParam>

    <ParamName>TYPE</ParamName>

    <ValEnum>VOICE,HOME</ValEnum>

    <ValEnum>FAX,HOME</ValEnum>

    <ValEnum>VOICE,CELL</ValEnum>

  </PropParam>

</Property>
```

5.3.23 Mod

Usage: Specifies the model name or model number of the device.

Parent Elements: `DevInf`

Restrictions: This element type is mandatory and must describe the external model identification as accurately as possible. It is not necessary to identify cosmetic or post production changes such as color or external logos that do not affect the operation of the device.

Content Model:

```
Mod (#PCDATA)
```

Attributes: None.

Example:

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

5.3.24 NoTruncate

Usage: Specifies if truncation is permitted should a property value exceed the maximum size as specified by the MaxSize tag. If the maximum size is exceeded the property MUST NOT be sent.

Parent Elements: `Property`

Restrictions: If the sending device has specified the <NoTruncate/> element in its Device Information for a particular field, then the receiving device MUST NOT (for server) or SHOULD NOT (for client) send a value for the field that has been, or would be truncated.

Content Model:

```
NoTruncate EMPTY
```

Attributes: None.

Example:

```
<Property>
  <PropName>TEL</PropName>
  <MaxSize>255</MaxSize>
  <NoTruncate/>
  <PropParam>
    <ParamName>TYPE</ParamName>
    <ValEnum>VOICE,HOME</ValEnum>
    <ValEnum>FAX,HOME</ValEnum>
    <ValEnum>VOICE,CELL</ValEnum>
  </PropParam>
</Property>
```

5.3.25 OEM

Usage: Specifies the OEM (Original Equipment Manufacturer) of the device.

Parent Elements: `DevInf`

Restrictions: This element type is optional.

Content Model:

```
OEM (#PCDATA)
```

Attributes: None.

Example:

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

5.3.26 ParamName

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

Parent Elements: `PropParam`

Restrictions: If the content type is either `text/x-vcard`, `text/vcard, text/x-vcalendar, text/calendar, application/vnd.omads-email+xml, application/vnd.omads-file+xml, or application/vnd.omads-folder+xml`, the value for this element type MUST be one of the values defined in the table below, or an extension value starting with ‘X’. Sending the ParamName element is optional if the device supports all the parameters of all the supported properties.

text/x-vcard	
PropName	**ParamName**
ADR	TYPE
EMAIL	TYPE
LABEL	TYPE
TEL	TYPE
PHOTO	TYPE
SOUND	TYPE
KEY	TYPE
LOGO	TYPE
text/x-vcalendar	
PropName	**ParamName**
ATTENDEE	ROLE, STATUS, RSVP, EXPECT
AALARM	TYPE
text/vcard	

PropName	**ParamName**
LOGO	TYPE
LABEL	TYPE
PHOTO	TYPE
ADR	TYPE
TEL	TYPE
EMAIL	TYPE
SOUND	TYPE
KEY	TYPE
text/calendar	
PropName	**ParamName**
ATTENDEE	CN , CUTYPE , DELEGATED-FROM , DELEGATED-TO , DIR , LANGUAGE , MEMBER , PARTSTAT , ROLE , RSVP , SENT-BY
ORGANIZER	CN , DIR , LANGUAGE , SENT-BY
application/vnd.omads-file	
PropName	**ParamName**
ATTRIBUTES	H, S, A, D, W, R, X
BODY	ENC
application/vnd.omads-email	
PropName	**ParamName**
EMAILITEM	ENC, TEXTTYPE, ATTACHTYPE
application/vnd.omads-folder	
PropName	**ParamName**
ATTRIBUTES	H, S, A, D , W, R, X

Content Model:

```
ParamName (#PCDATA)
```

Attributes: None.

Example: The following is an example of supporting both the CN and ROLE parameters of the vCalendar ATTENDEE property.

```
<Property>
  <PropName>ATTENDEE</PropName>
  <PropParam>
    <ParamName>CN</ParamName>
  </PropParam>
  <PropParam>
    <ParamName>ROLE</ParamName>
  </PropParam>
</Property>
```

5.3.27 Property

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

Parent Elements: `CTCap`

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

Content Model:

```
Property (PropName, DataType?, MaxOccur?, MaxSize?, NoTruncate?, ValEnum*,
DisplayName?, PropParam*)
```

Attributes: None.

Example:

```
<Property>
  <PropName>TEL</PropName>
  <MaxSize>255</MaxSize>
  <NoTruncate/>
  <MaxOccur>3</MaxOccur>
  <PropParam>
    <ParamName>TYPE</ParamName>
    <ValEnum>VOICE,HOME</ValEnum>
    <ValEnum>FAX,HOME</ValEnum>
    <ValEnum>VOICE,CELL</ValEnum>
  </PropParam>
</Property>
```

5.3.28 PropName

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

Parent Elements: `FilterCap, Property`

Restrictions: If the content type is either `text/x-vcard`, `text/vcard, text/x-vcalendar, text/calendar`, application/vnd.omads-email+xml, application/vnd.omads-file+xml, or application/vnd.omads-folder+xml, the value for this element type MUST be one of the values defined in the table below, or an extension value starting with 'X'.

Content type	**PropName**
text/x-vcard	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
text/x-vcalendar	BEGIN, VERSION, END, DAYLIGHT, GEO, PRODID, 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
text/vcard	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
text/calendar	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, PRODID, 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
application/vnd.omads-email	EMAIL, READ, FORWARD, REPLIED, RECEIVED, CREATED, MODIFIED, DELETED, FLAGGED, EMAILITEM, EXT, XNAM, XVAL
application/vnd.omads-file	FILE, NAME, CREATED, MODIFIED, ACCESSED, ATTRIBUTES, , CTTYPE, BODY, SIZE, EXT, XNAM, XVAL
application/vnd.omads-folder	FOLDER, NAME, CREATED, MODIFIED, ACCESSED, ATTRIBUTES,, ROLE, EXT, XNAM, XVAL

Content Model:

```
PropName (#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>
 <Property>
  <PropName>BEGIN</PropName>
  <ValEnum>VCALENDAR</ValEnum>
  <ValEnum>VEVENT</ValEnum>
 </Property>
 <Property>
  <PropName>VERSION</PropName>
  <ValEnum>1.0</ValEnum>
 </Property>
 <Property>
 <PropName>DTSTART</PropName>
</Property>
<Property>
 <PropName>DTEND</PropName>
</Property>
<Property>
 <PropName>DESCRIPTION</PropName>
</Property>
<Property>
 <PropName>END</PropName>
  <ValEnum>VCALENDAR</ValEnum>
  <ValEnum>VEVENT</ValEnum>
</Property>
</CTCap>
```

5.3.29 PropParam

Usage: Specifies a supported parameter of a given property

Parent Elements: `Property`

Restrictions: The content type capabilities of the device SHOULD be defined. If an enumeration of the possible parameter values is provided in ValEnum, then only these values or their combinations are allowed.

Content Model:

```
PropParam (ParamName, DataType?, ValEnum*, DisplayName?)
```

Attributes: None.

Example:

```
<PropParam>
  <ParamName>TYPE</ParamName>
  <ValEnum>VOICE, HOME</ValEnum>
  <ValEnum>FAX, HOME</ValEnum>
  <ValEnum>VOICE, CELL</ValEnum>
</PropParam>
```

5.3.30 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:

```
Rx (CTType, VerCT)
```

Attributes: None.

Example:

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

5.3.31 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:

```
Rx-Pref (CTType, VerCT)
```

Attributes: None.

Example:

```
<Rx-Pref>

  <CTType>text/vcard</CTType>

  <VerCT>3.0</VerCT>

</Rx-Pref>
```

5.3.32 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:

```
SharedMem EMPTY
```

Attributes: None.

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

```
<DSMem>

  <SharedMem/>

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

5.3.33 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:

```
SourceRef (#PCDATA)
```

Attributes: None.

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

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

5.3.34 SupportHierarchicalSync

Usage: Indicates the support for hierarchical sync. Note that this doesn't indicate the type of hierarchical contents that are supported.

Parent Elements: `DataStore`

Restrictions: This element MUST be present for a Datastore that supports hierarchical synchronisation. .

Content Model:

```
SupportHierarchicalSync EMPTY
```

Attributes: None.

Example:

```
<DevInf xmlns='syncml:devinf'>
   <VerDTD>1.2</VerDTD>
   <DevID>1218182THD012345-2</DevID>
   <DevTyp>pager</DevTyp>

   <DataStore>
      <SupportHierarchicalSync/>
   ...
   </DataStore>
</DevInf>
```

5.3.35 SupportLargeObjs

Usage: Specifies that the device supports receiving large objects.

Parent Element: `DevInf`

Restrictions: None

Content Model:

```
SupportLargeObjs EMPTY
```

Attributes: None.

Example:

```
<DevInf xmlns=’syncml:devinf’>

   <VerDTD>1.2</VerDTD>

   <Man>SomeManufacturer</Man>

   <Mod>SomeModel</Mod>

   <OEM>MyOEM</OEM>

   <FwV>1.0</FwV>

   <SwV>2.99</SwV>

   <HwV>2.1</HwV>

   <DevID>1218182THD012345-2</DevID>

   <DevTyp>pager</DevTyp>

   <SupportLargeObjs/>

   <DataStore>

   ...

   </DataStore>
</DevInf>
```

5.3.36 SupportNumberOfChanges

Usage: Specifies that the device supports number of changes

ParentElement: `DevInf`

Restrictions: Server SHOULD send <NumberOfChanges>, specified in Representation protocol specification v1.2 [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:

```
SupportNumberOfChanges EMPTY
```

Attributes: None.

Example:

```
<DevInf xmlns=’syncml:devinf’>

   <VerDTD>1.2</VerDTD>

   <Man>SomeManufacturer</Man>

   <Mod>SomeModel</Mod>

   <OEM>MyOEM</OEM>
```

```
    <FwV>1.0</FwV>
    <SwV>2.99</SwV>
    <HwV>2.1</HwV>
    <DevID>1218182THD012345-2</DevID>
    <DevTyp>pager</DevTyp>
    <SupportNumberOfChanges/>
    <DataStore>
    ...
    </DataStore>
</DevInf>
```

5.3.37 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. A software version can also have a date appended to it. If the content information includes 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 mandatory, and must uniquely identify the specific software build or version.

Content Model:

```
SwV (#PCDATA)
```

Attributes: None.

Example:

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

5.3.38 SyncCap

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

Parent Elements: DataStore

Restrictions: This element is mandatory.

Content Model:

```
SyncCap (SyncType+)
```

Attributes: None.

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

```
<DataStore>
...
  <SyncCap>
    <SyncType>1</SyncType>
    <SyncType>2</SyncType>
    <SyncType>7</SyncType>
  </SyncCap>
</DataStore>
```

5.3.39 SyncType

Usage: Specifies the type of the supported synchronization.

Parent Elements: SyncCap

Restrictions: If the device supports synchronizations specified in the OMA Data Sync protocol specification [**Error! Reference source not found.**], 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:

```
SyncType (#PCDATA)
```

Attributes: None.

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

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

5.3.40 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:

```
Tx (CTType, VerCT)
```

Attributes: None.

Example:

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

5.3.41 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:

```
Tx-Pref (CTType, VerCT)
```

Attributes: None.

Example:

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

5.3.42 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:

```
UTC EMPTY
```

Attributes: None.

Example:

```
<DevInf xmlns=’syncml:devinf’>

   <VerDTD>1.2</VerDTD>

   <Man>SomeManufacturer</Man>

   <Mod>SomeModel</Mod>

   <OEM>MyOEM</OEM>

   <FwV>1.0</FwV>

   <SwV>2.99</SwV>

   <HwV>2.1</HwV>

   <DevID>1218182THD012345-2</DevID>

   <DevTyp>pager</DevTyp>

   <UTC/>

   <DataStore>

   ...

   </DataStore>
</DevInf>
```

5.3.43 ValEnum

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

Parent Elements: `Property, PropParam`

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 or alternatively a comma-separated list of the values defined in the table below, or an extension value starting with ‘X’.

If a comma-separated list is used to specify possible values then:

- the ordering of values is irrelevant (e.g. VOICE, HOME is the same as HOME, VOICE).
- other than listed parameter combinations are not supported for Property. That is, a combination VOICE, CELL, HOME does not mean that also e.g. HOME and VOICE alone or HOME, CELL combination is supported.

Text/x-vcard	
PropName(;PropParam)	**ValEnum**

BEGIN	VCARD
END	VCARD
VERSION	2.1
ADR;TYPE	DOM, INTL, POSTAL, PARCEL, HOME, WORK
EMAIL;TYPE	AOL, AppleLink, ATTMail, CIS, eWorld, INTERNET, IBMMail, MCIMail, POWERSHARE, PRODIGY, TLX, X400
LABEL;TYPE	DOM, INTL, POSTAL, PARCEL, HOME, WORK
TEL;TYPE	PREF, WORK, HOME, VOICE, FAX, MSG, CELL, PAGER, BBS, MODEM, CAR, ISDN, VIDEO
PHOTO;TYPE	GIF, CGM, WMF, BMP, MET, PMB, DIB, PICT, TIFF, PS, PDF, JPEG, MPEG, MPEG2, AVI, QTIME
SOUND;TYPE	WAVE, PCM, AIFF
KEY;TYPE	X509, PGP
LOGO;TYPE	GIF, CGM, WMF, BMP, MET, PMB, DIB, PICT, TIFF, PS, PDF, JPEG, MPEG, MPEG2, AVI, QTIME
Text/x-vcalendar	
PropName(;PropParam)	**ValEnum**
BEGIN	VCALENDAR, VEVENT, VTODO
END	VCALENDAR, VEVENT, VTODO
VERSION	1.0
CLASS	PUBLIC, PRIVATE, CONFIDENTIAL
AALARM;TYPE	WAVE, PCM, AIFF
Text/vcard	
PropName(;PropParam)	**ValEnum**
BEGIN	VCARD
END	VCARD
VERSION	3.0
CLASS	PUBLIC, PRIVATE, CONFIDENTIAL

LOGO;TYPE	GIF, CGM, WMF, BMP, MET, PMB, DIB, PICT, TIFF, PS, PDF, JPEG, MPEG, MPEG2, AVI, QTIME, other IANA registered image types
LABEL;TYPE	DOM, INTL, POSTAL, PARCEL, HOME, WORK, other IANA registered parameter names
PHOTO;TYPE	GIF, CGM, WMF, BMP, MET, PMB, DIB, PICT, TIFF, PS, PDF, JPEG, MPEG, MPEG2, AVI, QTIME, other IANA registered image types
ADR;TYPE	DOM, INTL, POSTAL, PARCEL, HOME, WORK, other IANA registered parameter names
TEL;TYPE	PREF, WORK, HOME, VOICE, FAX, MSG, CELL, PAGER, BBS, MODEM, CAR, ISDN, VIDEO, other IANA registered parameter names
EMAIL;TYPE	PREF, INTERNET, X400, other IANA registered address types
SOUND;TYPE	WAVE, PCM, AIFF, other IANA registered audio formats
KEY;TYPE	X509, PGP, other IANA registered certificate types
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:

```
ValEnum (#PCDATA)
```

Attributes: None.

Example: The following is an example of supporting iCalendar binary attachments.

```
<CTCap>

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

5.3.44 VerCT

Usage: Specifies the version of a supported content type.

Parent Elements: `CTCap, Filter-Rx, FilterCap, Rx, Rx-Pref, Tx, Tx-Pref`

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

Content Model:

```
VerCT (#PCDATA)
```

Attributes: None.

Example:

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

5.3.45 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 reversions of the specification create incompatible changes that will generally require a new parser. Minor revisions involve changes that do not impact basic compatibility of the parser. When the XML document conforms to this revision of the Device Information specification the value MUST be 1.2. The element type MUST be included in the DevInf element.

Content Model:

```
VerDTD (#PCDATA)
```

Attributes: None.

Example:

```
<DevInf xmlns=’syncml:devinf’>
  <VerDTD>1.2</VerDTD>

  ...

</DevInf>
```

5.3.46 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:

```
XNam (#PCDATA)
```

Attributes: None.

Example:

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

5.3.47 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:

```
XVal (#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) V1.2 Document Type Definition
modified 09 Jan 2004

Copyright Open Mobile Alliance Ltd., 2002-2004
          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 1.2//EN"
             "http://www.openmobilealliance.org/tech/DTD/OMA-SyncML-
Device_Information-DTD-1.2.dtd"
             [<?oma-syncml-devinf-ver supported-versions="1.2"?>]>
   <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,
UTC?,SupportLargeObjs?, SupportNumberOfChanges?,DataStore+, Ext*)>
<!ELEMENT VerDTD (#PCDATA)>
<!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 UTC EMPTY>
<!ELEMENT SupportLargeObjs EMPTY>
<!ELEMENT SupportNumberOfChanges EMPTY>
<!ELEMENT DataStore (SourceRef, DisplayName?, MaxGUIDSize?, Rx-Pref, Rx*,
Tx-Pref, Tx*, CTCap+, DSMem?, SupportHierarchicalSync?, SyncCap, Filter-
Rx*, FilterCap*)>
<!ELEMENT SourceRef (#PCDATA)>
<!ELEMENT DisplayName (#PCDATA)>
<!ELEMENT MaxGUIDSize (#PCDATA)>
<!ELEMENT Rx-Pref (CTType, VerCT)>
<!ELEMENT Rx (CTType, VerCT)>
<!ELEMENT Tx-Pref (CTType, VerCT)>
<!ELEMENT Tx (CTType, VerCT)>
<!ELEMENT Filter-Rx (CTType, VerCT)>
<!ELEMENT FilterCap (CTType, VerCT, FilterKeyword*, PropName*)>
<!ELEMENT FilterKeyword (#PCDATA)>
<!ELEMENT CTCap (CTType, VerCT, FieldLevel?, Property+)>
<!ELEMENT CTType (#PCDATA)>
<!ELEMENT VerCT (#PCDATA)>
<!ELEMENT FieldLevel EMPTY>
<!ELEMENT Property (PropName, DataType?, MaxOccur?, MaxSize?, NoTruncate?,
ValEnum*, DisplayName?, PropParam*)>
<!ELEMENT PropName (#PCDATA)>
<!ELEMENT DataType (#PCDATA)>
<!ELEMENT MaxOccur (#PCDATA)>
<!ELEMENT MaxSize (#PCDATA)>
<!ELEMENT NoTruncate EMPTY>
<!ELEMENT ValEnum (#PCDATA)>
<!ELEMENT PropParam (ParamName, DataType?, ValEnum*, DisplayName?)>
<!ELEMENT ParamName (#PCDATA)>
<!ELEMENT DSMem (SharedMem?, MaxMem?, MaxID?)>
<!ELEMENT MaxMem (#PCDATA)>
<!ELEMENT MaxID (#PCDATA)>
<!ELEMENT SupportHierarchicalSync EMPTY>
<!ELEMENT SyncCap (SyncType+)>
<!ELEMENT SyncType (#PCDATA)>
<!ELEMENT Ext (XNam,XVal*)>
<!ELEMENT XNam (#PCDATA)>
<!ELEMENT XVal (#PCDATA)>
<!ELEMENT SharedMem EMPTY>
<!--End of DTD -->
```

7. WBXML Definitions

This version of the DevInf DTD specification is associated with the WBXML code space 0x1203 and the formal public identifier `-//SYNCML//DTD DevInf 1.2//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
MaxSize	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
Property	2B
PropParam	2C
MaxOccur	2D
NoTruncate	2E
Filter-Rx	30
FilterCap	31
FilterKeyword	32
FieldLevel	33
SupportHierarchicalSync	34

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>1.2</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>

    <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/vcard</CTType>
      <VerCT>2.1</VerCT>
    </Tx>

    <CTCap>
      <CTType>text/vcard</CTType>
      <VerCT>3.0</VerCT>

      <Property>
        <PropName>BEGIN</PropName>
        <ValEnum>VCARD</ValEnum>
      </Property>

      <Property>
```

```
        <PropName>END</PropName>
        <ValEnum>VCARD</ValEnum>
      </Property>

      <Property>
        <PropName>VERSION</PropName>
        <ValEnum>3.0</ValEnum>
      </Property>

      <Property>
        <PropName>N</PropName>
      </Property>

      <Property>
        <PropName>TEL</PropName>
        <PropParam>
          <ParamName>TYPE</ParamName>
          <ValEnum>VOICE,HOME</ValEnum>
          <ValEnum>FAX,HOME</ValEnum>
          <ValEnum>VOICE,CELL</ValEnum>
        </PropParam>
      </Property>

    </CTCap>

    <CTCap>
      <CTType>text/x-vcard</CTType>
      <VerCT>2.1</VerCT>

      <Property>
        <PropName>BEGIN</PropName>
        <ValEnum>VCARD</ValEnum>
      </Property>

      <Property>
        <PropName>END</PropName>
        <ValEnum>VCARD</ValEnum>
      </Property>

      <Property>
        <PropName>VERSION</PropName>
        <ValEnum>2.1</ValEnum>
      </Property>

      <Property>
        <PropName>N</PropName>
      </Property>

      <Property>
        <PropName>TEL</PropName>
        <PropParam>
          <ParamName>TYPE</ParamName>
          <ValEnum>VOICE,HOME</ValEnum>
          <ValEnum>FAX,HOME</ValEnum>
          <ValEnum>VOICE,CELL</ValEnum>
        </PropParam>
      </Property>

    </CTCap>

    <DSMem>
      <MaxMem>32650</MaxMem>
      <MaxID>250</MaxID>
    </DSMem>

    <SyncCap>
      <SyncType>1</SyncType>
      <SyncType>2</SyncType>
      <SyncType>7</SyncType>
    </SyncCap>

  </DataStore>

  <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: mailto: admins@syncml.org

Intended usage: COMMON

Author/Change controller: mailto: 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.

```
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): BDM

Macintosh File Type Code(s): BDML

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

Intended usage: COMMON

Author/Change controller: mailto: admins@syncml.org

Appendix A. Static Conformance Requirements (Normative)

A.1 Client Device Information

Table 1 – Client Device Information Elements

Item	Functionality	Reference	Status	Requirement
SCR-DS-DEVINF-C-001	Support for CTCap element	5.3.1	O	SCR-DS-DEVINF-C-002 AND SCR-DS-DEVINF-C-038 AND SCR-DS-DEVINF-C-042
SCR-DS-DEVINF-C-002	Support for CTType element	5.3.2	M	
SCR-DS-DEVINF-C-003	Support for DataStore element	5.3.3	M	SCR-DS-DEVINF-C-001 AND SCR-DS-DEVINF-C-018 AND SCR-DS-DEVINF-C-025 AND SCR-DS-DEVINF-C-026 AND SCR-DS-DEVINF-C-028 AND SCR-DS-DEVINF-C-032 AND SCR-DS-DEVINF-C-034 AND SCR-DS-DEVINF-C-035
SCR-DS-DEVINF-C-004	Support for DataType element	5.3.4	O	
SCR-DS-DEVINF-C-005	Support for DevID element	5.3.5	M	
SCR-DS-DEVINF-C-006	Support for DevInf element	5.3.6	M	SCR-DS-DEVINF-C-003 AND SCR-DS-DEVINF-C-005 AND SCR-DS-DEVINF-C-007 AND SCR-DS-DEVINF-C-015 AND SCR-DS-DEVINF-C-016 AND SCR-DS-DEVINF-C-017 AND SCR-DS-DEVINF-C-021 AND SCR-DS-DEVINF-C-031 AND SCR-DS-DEVINF-C-039
SCR-DS-DEVINF-C-007	Support for DevTyp element	5.3.7	M	
SCR-DS-DEVINF-C-008	Support for DisplayName element	5.3.8	O	
SCR-DS-DEVINF-C-009	Support for DSMem element	5.3.9	O	SCR-DS-DEVINF-C-019 OR SCR-DS-DEVINF-C-020 OR SCR-DS-DEVINF-C-027
SCR-DS-DEVINF-C-010	Support for Ext element	5.3.10	O	SCR-DS-DEVINF-C-040 AND SCR-DS-DEVINF-C-041
SCR-DS-DEVINF-C-011	Support for FieldLevel element	5.3.11	O	
SCR-DS-DEVINF-C-012	Support for Filter-Rx element	5.3.12	O	SCR-DS-DEVINF-C-002 AND SCR-DS-DEVINF-C-038
SCR-DS-DEVINF-C-013	Support for FilterCap element	5.3.13	O	SCR-DS-DEVINF-C-002 AND SCR-DS-DEVINF-C-038 AND (SCR-DS-DEVINF-C-014 OR

				SCR-DS-DEVINF-C-024)
SCR-DS-DEVINF-C-014	Support for FilterKeyword element	5.3.14	O	
SCR-DS-DEVINF-C-015	Support for FwV element	5.3.15	M	
SCR-DS-DEVINF-C-016	Support for HwV element	5.3.16	M	
SCR-DS-DEVINF-C-017	Support for Man element	5.3.17	M	
SCR-DS-DEVINF-C-018	Support for MaxGUIDSize element	5.3.18	M	
SCR-DS-DEVINF-C-019	Support for MaxID element	5.3.19	O	
SCR-DS-DEVINF-C-020	Support for MaxMem element	5.3.20	O	
SCR-DS-DEVINF-C-021	Support for Mod element	5.3.23	M	
SCR-DS-DEVINF-C-022	Support for OEM element	5.3.25	O	
SCR-DS-DEVINF-C-023	Support for ParamName element	5.3.26	O	
SCR-DS-DEVINF-C-024	Support for PropName element	5.3.28	M	
SCR-DS-DEVINF-C-025	Support for Rx element	5.3.30	M	SCR-DS-DEVINF-C-002 AND SCR-DS-DEVINF-C-038
SCR-DS-DEVINF-C-026	Support for Rx-Pref element	5.3.31	M	SCR-DS-DEVINF-C-002 AND SCR-DS-DEVINF-C-038
SCR-DS-DEVINF-C-027	Support for SharedMem element	5.3.32	O	
SCR-DS-DEVINF-C-028	Support for SourceRef element	5.3.33	M	
SCR-DS-DEVINF-C-029	Include SupportLargeObjs element in the DevInf	5.3.35	O	
SCR-DS-DEVINF-C-030	Support for SupportNumberOfChanges element	5.3.35	O	
SCR-DS-DEVINF-C-031	Support for SwV element	5.3.36	M	
SCR-DS-DEVINF-C-032	Support for SyncCap element	5.3.37	M	SCR-DS-DEVINF-C-033
SCR-DS-DEVINF-C-033	Support for SyncType element	5.3.38	M	
SCR-DS-DEVINF-C-034	Support for Tx element	5.3.39	M	SCR-DS-DEVINF-C-002 AND SCR-DS-DEVINF-C-038
SCR-DS-DEVINF-C-035	Support for Tx-Pref element	5.3.40	M	SCR-DS-DEVINF-C-002 AND SCR-DS-DEVINF-C-038
SCR-DS-DEVINF-C-036	Support for UTC element	5.3.41	O	

SCR-DS-DEVINF-C-037	Support for ValEnum element	5.3.42	M	
SCR-DS-DEVINF-C-038	Support for VerCT element	5.3.43	M	
SCR-DS-DEVINF-C-039	Support for VerDTD element	5.3.44	M	
SCR-DS-DEVINF-C-040	Support for Xnam element	5.3.45	O	
SCR-DS-DEVINF-C-041	Support for Xval element	5.3.46	O	
SCR-DS-DEVINF-C-042	Support for Property element	5.3.27	M	SCR-DS-DEVINF-C-024 AND SCR-DS-DEVINF-C-037 AND SCR-DS-DEVINF-C-043
SCR-DS-DEVINF-C-043	Support for PropParam element	5.3.29	M	SCR-DS-DEVINF-C-023 AND SCR-DS-DEVINF-C-037
SCR-DS-DEVINF-C-044	Support for MaxOccur element	5.3.21	O	
SCR-DS-DEVINF-C-045	Support for MaxSize element	5.3.22	O	
SCR-DS-DEVINF-C-046	Support for NoTruncate element	5.3.24	O	

A.2 Server Device Information

Table 2 – Server Device Information Elements

Item	Functionality	Reference	Status	Requirement
SCR-DS-DEVINF-S-001	Support for CTCap element	5.3.1	O	SCR-DS-DEVINF-S-002 AND SCR-DS-DEVINF-S-038 AND SCR-DS-DEVINF-S-042
SCR-DS-DEVINF-S-002	Support for CTType element	5.3.2	M	
SCR-DS-DEVINF-S-003	Support for DataStore element	5.3.3	M	SCR-DS-DEVINF-S-001 AND SCR-DS-DEVINF-S-018 AND SCR-DS-DEVINF-S-025 AND SCR-DS-DEVINF-S-026 AND SCR-DS-DEVINF-S-028 AND SCR-DS-DEVINF-S-032 AND SCR-DS-DEVINF-S-034 AND SCR-DS-DEVINF-S-035
SCR-DS-DEVINF-S-004	Support for DataType element	5.3.4	M	
SCR-DS-DEVINF-S-005	Support for DevID element	5.3.5	M	
SCR-DS-DEVINF-S-006	Support for DevInf element	5.3.6	M	SCR-DS-DEVINF-S-003 AND SCR-DS-DEVINF-S-005 AND SCR-DS-DEVINF-S-007 AND SCR-DS-DEVINF-S-015 AND SCR-DS-DEVINF-S-016 AND SCR-DS-DEVINF-S-017 AND SCR-DS-DEVINF-S-021 AND SCR-DS-DEVINF-S-029 AND SCR-DS-DEVINF-S-030 AND

				SCR-DS-DEVINF-S-031 AND SCR-DS-DEVINF-S-036 AND SCR-DS-DEVINF-S-039
SCR-DS-DEVINF-S-007	Support for DevTyp element	5.3.7	M	
SCR-DS-DEVINF-S-008	Support for DisplayName element	5.3.8	O	
SCR-DS-DEVINF-S-009	Support for DSMem element	5.3.9	O	SCR-DS-DEVINF-S-019 OR SCR-DS-DEVINF-S-020 OR SCR-DS-DEVINF-S-027
SCR-DS-DEVINF-S-010	Support for Ext element	5.3.10	O	SCR-DS-DEVINF-S-040 AND SCR-DS-DEVINF-S-041
SCR-DS-DEVINF-S-011	Support for FieldLevel element	5.3.11	O	
SCR-DS-DEVINF-S-012	Support for Filter-Rx element	5.3.12	O	SCR-DS-DEVINF-S-002 AND SCR-DS-DEVINF-S-038
SCR-DS-DEVINF-S-013	Support for FilterCap element	5.3.13	O	SCR-DS-DEVINF-S-002 AND SCR-DS-DEVINF-S-038 AND (SCR-DS-DEVINF-S-014 OR SCR-DS-DEVINF-S-024)
SCR-DS-DEVINF-S-014	Support for FilterKeyword element	5.3.14	O	
SCR-DS-DEVINF-S-015	Support for FwV element	5.3.15	M	
SCR-DS-DEVINF-S-016	Support for HwV element	5.3.16	M	
SCR-DS-DEVINF-S-017	Support for Man element	5.3.17	M	
SCR-DS-DEVINF-S-018	Support for MaxGUIDSize element	5.3.18	M	
SCR-DS-DEVINF-S-019	Support for MaxID element	5.3.19	O	
SCR-DS-DEVINF-S-020	Support for MaxMem element	5.3.20	O	
SCR-DS-DEVINF-S-021	Support for Mod element	5.3.23	M	
SCR-DS-DEVINF-S-022	Support for OEM element	5.3.25	O	
SCR-DS-DEVINF-S-023	Support for ParamName element	5.3.26	M	
SCR-DS-DEVINF-S-024	Support for PropName element	5.3.28	M	
SCR-DS-DEVINF-S-025	Support for Rx element	5.3.30	M	SCR-DS-DEVINF-S-002 AND SCR-DS-DEVINF-S-038
SCR-DS-DEVINF-S-026	Support for Rx-Pref element	5.3.31	M	SCR-DS-DEVINF-S-002 AND SCR-DS-DEVINF-S-038
SCR-DS-DEVINF-S-027	Support for SharedMem	5.3.32	O	

	element			
SCR-DS-DEVINF-S-028	Support for SourceRef element	5.3.33	M	
SCR-DS-DEVINF-S-029	Include SupportLargeObjs element in the DevInf	5.3.35	M	
SCR-DS-DEVINF-S-030	Support for SupportNumberOfChanges element	5.3.35	M	
SCR-DS-DEVINF-S-031	Support for SwV element	5.3.36	M	
SCR-DS-DEVINF-S-032	Support for SyncCap element	5.3.37	M	SCR-DS-DEVINF-S-033
SCR-DS-DEVINF-S-033	Support for SyncType element	5.3.38	M	
SCR-DS-DEVINF-S-034	Support for Tx element	5.3.39	M	SCR-DS-DEVINF-S-002 AND SCR-DS-DEVINF-S-038
SCR-DS-DEVINF-S-035	Support for Tx-Pref element	5.3.40	M	SCR-DS-DEVINF-S-002 AND SCR-DS-DEVINF-S-038
SCR-DS-DEVINF-S-036	Support for UTC element	5.3.41	M	
SCR-DS-DEVINF-S-037	Support for ValEnum element	5.3.42	M	
SCR-DS-DEVINF-S-038	Support for VerCT element	5.3.43	M	
SCR-DS-DEVINF-S-039	Support for VerDTD element	5.3.44	M	
SCR-DS-DEVINF-S-040	Support for Xnam element	5.3.45	O	
SCR-DS-DEVINF-S-041	Support for Xval element	5.3.46	O	
SCR-DS-DEVINF-S-042	Support for Property element	5.3.27	M	SCR-DS-DEVINF-S-004 AND SCR-DS-DEVINF-S-024 AND SCR-DS-DEVINF-S-037 AND SCR-DS-DEVINF-S-043 AND SCR-DS-DEVINF-S-044 AND SCR-DS-DEVINF-S-045 AND SCR-DS-DEVINF-S-046
SCR-DS-DEVINF-S-043	Support for PropParam element	5.3.29	M	SCR-DS-DEVINF-S-004 AND SCR-DS-DEVINF-S-023 AND SCR-DS-DEVINF-S-037
SCR-DS-DEVINF-S-044	Support for MaxOccur element	5.3.21	M	
SCR-DS-DEVINF-S-045	Support for MaxSize element	5.3.22	M	
SCR-DS-DEVINF-S-046	Support for NoTruncate element	5.3.24	M	

Appendix B. Change History (Informative)

B.1 Approved Version History

Reference	Date	Description
OMA-SyncML-DevInfo-V1_1_2-20030612-A	12 Jun 2003	Initial OMA release
OMA-TS-DS_DevInf-V1_2-20060710-A	10 Jul 2006	Approved by TP ref#OMA-TP-2006-0239R03-INP_DS_V1_2_for_final_approval