OMA DS Device Information

Approved Version 2.0 – 19 Jul 2011

Open Mobile Alliance

OMA-TS-DS-DevInf-V2_0-20110719-A

Contents

TABLES

1. Scope

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

Please refer to [DSCONCEPTS] for further information on the OMA DS organization and history.

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

[DSCONCEPTS] "Data Synchronization Concepts and Definitions", Open Mobile Alliance™, OMA-TS-DS_Concepts-V2_0, URL:http://www.openmobilealliance.org/

[DSHISTORY] "OMA DS Standards Change History", Open Mobile Alliance™, OMA-WP-SyncML_ChangeHistory, URL:http://www.openmobilealliance.org/

[DSPRO] "Data Synchronization Protocol", Open Mobile Alliance™, OMA-TS-DS_Protocol-V2_0, URL:http://www.openmobilealliance.org/

[DSSYNTAX] "Data Synchronization Syntax", Open Mobile Alliance™, OMA-TS-DS_Syntax-V2_0, URL:http://www.openmobilealliance.org/

[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

[RFC2119] "Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997, URL:http://www.ietf.org/rfc/rfc2119.txt

[RFC2279] "UTF-8, a transformation format of ISO 10646", F. Yergeau, January 1998, URL:http://www.ietf.org/rfc/rfc2279.txt

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

[XML] "Extensible Markup Language (XML) 1.0", World Wide Web Consortium Recommendation, http://www.w3.org/TR/REC-xml

2.2 Informative References

None.

Please refer to [DSCONCEPTS] for the other Informative References.

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 [RFC2119].

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

Any reference to components of the Data Synchronization XML Schema or XML snippets is specified in this `typeface`.

3.2 Definitions

Please refer to the [DSCONCEPTS] document.

3.3 Abbreviations

Please refer to the [DSCONCEPTS] document.

4. Introduction

This document defines the Schema for the XML representation of the Device Information (DevInf Schema). The DevInf Schema 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.

4.1 Version History

For a detailed change history of OMA-DS, refer to **[DSHISTORY]**.

Specific Device Information changes include:

4.1.1 Version 1.0.1

`CTCap` was moved from the Syntax to the Device Information, and *`Version`* (named `VerDTD` at the time) was added.

4.1.2 Version 1.1

`UTC`, `SupportNumberOfChanges`, and `SupportLargeObjs` were added.

4.1.3 Version 1.2

`Size` was renamed to `MaxSize`. `Property`, `PropParam`, `MaxOccur`, *`Truncate`* (named `NoTruncate` at the time), `Filter-Rx`, `FilterCap`, `FilterKeyword`, `FieldLevel`, and *`SupportHierarchicalSync`* were added. *`Version`* (named `VerDTD` at the time) was added.

4.1.4 Version 2.0

`DevCap`, `ExtURI`, *`FPUnique`*, *`MaxStoredAnchors`*, *`SupportAtomic`*, *`SupportEncryption`*, *`SupportFieldLevel`*, *`SupportSequence`*, *`SupportSftDel`*, `PropInfo`, `RxTx-CT`, `StoredAnchors`, and `ValidAnchor` were added. `DevTyp` was renamed to `DevType`. `XNam` was renamed to `XName`. `XVal` was renamed to `XValue`. `VerDTD` was renamed to *`Version`*. The flags and simple values of *`DisplayName`*, *`FieldLevel`*, *`FPUnique`*, *`MaxGUIDSize`*, *`MaxID`*, *`MaxMem`*, *`MaxStoredAnchors`*, *`ParamName`*, *`SharedMem`*, *`SupportAtomic`*, *`SupportEncryption`*, *`SupportFieldLevel`*, *`SupportHierarchicalSync`*, *`SupportLargeObjs`*, *`SupportNumberOfChanges`*, *`SupportSequence`*, *`SupportSftDel`*, *`Truncate`*, *`UTC`*, and *`Version`* were converted from elements to attributes.

5. Device Information

5.1 XML Usage

The device information is represented in a mark up language defined by [XML]. The Device Information Schema defines the XML schema used to represent information about the capabilities of a data synchronization device.

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

```
-//SyncML//Schema DevInf 2.0//EN
```

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

```
devinf20
```

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

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, schema references, and namespaces. 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 following examples should be considered equivalent:

Fully Qualified: This style is both well-formed and valid (provided a copy of the DevInf schema is available). This style is recommended for file formats, such as storing device information for later use, and SHOULD be used for external URI references.

```
<?xml version="1.0" encoding="UTF-8"?>
<DevInf xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="OMA-SUP-XSD_DS_DevInf_Schema-V2_0.xsd"
    Version="2.0">
  ...

</DevInf>
```

Minimally Qualified: This style uniquely identifies the namespace as being OMA-DS Device Information, and is well-formed, but not valid. This style SHOULD be used when Device Information is included in an OMA-DS message.

```
<DevInf xmlns='syncml:devinf' Version="2.0">
  ...
</DevInf>
```

Unqualified: This style MAY be used where the context is clear through other information.

```
<DevInf Version="2.0">
  ...
</DevInf>
```

Namespace Qualified: This style MAY be used where Device Information is included inside other XML documents, and the appropriate namespaces are declared. This style MUST be both well-formed and valid. The choice of namespaces (for elements, and attributes) is arbitrary. The use of any style like this SHOULD NOT be used for over-the-air communications.

```
<devinf:DevInf devinfA:Version="2.0">
  ...
</devinf:DevInf>
```

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 process ability. In addition, XML allows the originator to capture the structure of a document, not just its 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 WBXML Usage

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 8 of this document. The use of [WBXML] format is external to this specification and should be transparent to any XML application supporting this Schema. 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.

For the purposes of OMA-DS, WBXML 1.1, WBXML 1.2 and WBXML 1.3 are functionally equivalent, and all MUST be accepted in implementations that support WBXML. Effectively, this merely requires the WBXML parser to accept 01, 02 or 03 as the first byte of the document.

5.3 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.

5.4 Default Values

Various settings in the Device Information document have default values that apply if not specified. These default values were chosen to be representative of the minimum set of features for a simple, but reasonably efficient client, such that the

client would only have to specify a minimum number of settings. Clients MAY override the default values, including removing support for functionality, but are cautioned to consider those choices carefully.

The features that are optional, but should generally be supported include:

- FPUnique
- Truncate
- SupportLargeObjs
- SupportNumberOfChanges

5.5 Security Considerations

Security considerations for Device Information are generally the same as the OMA-DS messages that they are contained in, and are addressed by the methods contained in [DSPRO].

One Device Information specific concern is to allow for receiving different device information from devices that otherwise appear to be of the same type. This would prevent a modified device implementation from sending different device information (or a new device with old identifiers), and having that information be used on inappropriate devices. In other words, device information cannot always be assumed to be the same within an entire class of devices, and the device information from a specific device may vary over time.

6. Device Information Markup Language

6.1 Common Type for Device Information Elements

The following common type is defined for the Device Information Elements.

6.1.1 RxTxType

Usage: This type definition is used in the Device Information Elements definition. It is used to specify the content type capabilities.

Used in Elements: `Filter-Rx`, `Rx`, `Rx-Pref`, `Tx`, `Tx-Pref`

Content Model:

```
<xs:complexType name="RxTxType">
  <xs:sequence>
    <xs:element ref="CTType"/>
    <xs:element ref="VerCT" "/>
  </xs:sequence>
</xs:complexType>
```

Example:

```
<xs:element name="Rx" type="RxTxType" />
```

6.2 Device Information Element Descriptions

The following element and attribute types are included in the Device Information Schema.

6.2.1 CTCap

Usage: Specifies the content type capabilities of the specific data store.

Parent Element: `DataStore`

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

Content Model:

```
<xs:element name="CTCap" type="CTCapType" />

<xs:complexType name="CTCapType">
  <xs:sequence>
    <xs:element ref="CTType" />
    <xs:element ref="VerCT" />
    <xs:element ref="Property" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute ref="FieldLevel" use="optional"/>
</xs:complexType>
```

Attributes: *`FieldLevel`*

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">
      <ValEnum>VOICE,HOME</ValEnum>
      <ValEnum>FAX,HOME</ValEnum>
      <ValEnum>VOICE,CELL</ValEnum>
    </PropParam>
  </Property>
</CTCap>
```

6.2.2 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 [DSSYNTAX] . Other values can also be specified.

Content Model:

```
<xs:element name="CTType" type="xs:string"/>
```

Attributes: None.

Example:

```
<CTCap>
  <CTType>text/vcard</CTType>
  ...
</CTCap>
```

6.2.3 DataStore

Usage: Specifies the properties of a given local datastore.

Parent Element: `DevInf`

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

Content Model:

```
<xs:element name="DataStore" type="DataStoreType" />

<xs:complexType name="DataStoreType">
  <xs:sequence>
    <xs:element ref="SourceRef" />
    <xs:element ref="RxTx-CT" />
    <xs:element ref="CTCap" maxOccurs="unbounded"/>
    <xs:element ref="DSMem" minOccurs="0"/>
    <xs:element ref="SyncCap" />
    <xs:element ref="Filter-Rx" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="FilterCap" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute ref="DisplayName" use="optional"/>
  <xs:attribute ref="MaxGUIDSize" use="optional"/>
</xs:complexType>
```

Attributes: *DisplayName*, *MaxGUIDSize*

Example:

```
<DataStore DisplayName="Addressbook" MaxGUIDSize="32">
  <SourceRef>contacts</SourceRef>
  <RxTx-CT>
    ... ...
  </RxTx-CT>
  <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">
        <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">
        <ValEnum>VOICE,HOME</ValEnum>
        <ValEnum>FAX,HOME</ValEnum>
        <ValEnum>VOICE,CELL</ValEnum>
      </PropParam>
    </Property>
  </CTCap>
  <SyncCap FPUnique="true"/>
  <DSMem SharedMem="true" MaxMem="65539" MaxID="512" />
</DataStore>
```

6.2.4 DataType

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

Parent Elements: `PropInfo`, `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:

```
<xs:element name="DataType" type="xs:string" />
```

Attributes: None.

Example:

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

6.2.5 DevCap

Usage: The placeholder element for some specific device capability elements.

Parent Elements: `DevInf`

Restrictions: This element is mandatory.

Content Model:

```
<xs:element name="DevCap" type="DevCapType"/>

<xs:complexType name="DevCapType">
      <xs:sequence>
        <xs:element ref="Man" />
        <xs:element ref="Model" />
        <xs:element ref="OEM" minOccurs="0"/>
        <xs:element ref="FwV" />
        <xs:element ref="SwV" />
        <xs:element ref="HwV" />
        <xs:element ref="DevID" />
        <xs:element ref="DevType" />
      </xs:sequence>
  <xs:attribute ref="SupportAtomic" use="optional"/>
  <xs:attribute ref="SupportEncryption" use="optional"/>
  <xs:attribute ref="SupportFieldLevel" use="optional"/>
  <xs:attribute ref="SupportLargeObjs" use="optional"/>
  <xs:attribute ref="SupportNumberOfChanges" use="optional"/>
  <xs:attribute ref="SupportSftDel" use="optional"/>
  <xs:attribute ref="SupportSequence" use="optional"/>
  <xs:attribute ref="UTC" use="optional"/>
</xs:complexType>
```

Attributes: *SupportAtomic*, *SupportEncryption*, *SupportFieldLevel*, *SupportLargeObjs*, *SupportNumberOfChanges*, *SupportSequence*, *UTC*

Example:

```
<DevCap UTC="false">
  <Man>SomeManufacturer</Man>
  <Model>SomeModel</Model>
  <OEM>MyOEM</OEM>
  <FwV>1.0</FwV>
  <SwV>2.99</SwV>
  <HwV>2.1</HwV>
  <DevID>1218182THD012345-2</DevID>
  <DevType>pager</DevType>
</DevCap>
```

6.2.6 DevID

Usage: Specifies the identifier of the source synchronization device.

Parent Element: `DevCap`

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 DS Notification.

Content Model:

```
<xs:element name="DevID" type="xs:string" />
```

Attributes: None.

Example:

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

6.2.7 DevInf

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

Parent Element: None.

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

Content Model:

```
<xs:element name="DevInf" type="DevInfType">
  <xs:annotation>
    <xs:documentation>Root</xs:documentation>
  </xs:annotation>
</xs:element>

<xs:complexType name="DevInfType">
  <xs:sequence>
  <xs:element ref="ExtURI" minOccurs="0"/>
    <xs:element ref="DevCap" />
    <xs:element ref="DataStore" maxOccurs="unbounded"/>
    <xs:element ref="Ext" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute ref="Version" use="required"/>
</xs:complexType>
```

Attributes:

Name	Type	Occurrence	Description
xmlns	CDATA	IMPLICIT	SHOULD be present with a value of **'syncml:devinf'** for Device Information included in an OMA-DS Message. See section 5.1

Example:

```
<DevInf xmlns='syncml:devinf' Version="2.0">
  <DevCap>
    <Man>SomeManufacturer</Man>
    <Model>SomeModel</Model>
    <OEM>MyOEM</OEM>
    <FwV>1.0</FwV>
    <SwV>2.99</SwV>
    <HwV>2.1</HwV>
    <DevID>1218182THD012345-2</DevID>
    <DevType>pager</DevType>
  </DevCap>
  <DataStore>
   ...
  </DataStore>
</DevInf>
```

6.2.8 DevType

Usage: Specifies the type of the source synchronization device.

Parent Element: `DevCap`

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.

DevType	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:

```
<xs:element name="DevType" type="xs:string" />
```

Attributes: None.

Example:

```
<DevType>pager</DevType>
```

6.2.9 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`, `PropInfo`, `PropParam`

Restrictions: This attribute is optional.

Content Model:

```
<xs:attribute name="DisplayName" type="xs:string" />
```

Attributes: None.

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

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

6.2.10 DSMem

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

Parent Element: `DataStore`

Restrictions: The element type is optional.

Content Model:

```
<xs:element name="DSMem" type="DSMemType" />
<xs:complexType name="DSMemType">
  <xs:attribute ref="SharedMem" use="optional"/>
  <xs:attribute ref="MaxMem" use="optional"/>
  <xs:attribute ref="MaxID" use="optional"/>
</xs:complexType>
```

Attributes: *`MaxID`*, *`MaxMem`*, *`SharedMem`*

Example: The following example specifies a shared datastore memory.

```
<DSMem SharedMem="true" MaxMem="65539" MaxID="512"/>
```

6.2.11 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 Element: `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:

```
<xs:element name="Ext" type="ExtType" />

<xs:complexType name="ExtType">
  <xs:sequence>
    <xs:element ref="XName" />
    <xs:element ref="XValue" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

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>
  <XName>X-Foo-CliVer</XName>
  <XValue>5.0</XValue>
  <XValue>5.01</XValue>
  <XValue>5.02</XValue>
</Ext>
```

6.2.12 ExtURI

Usage: Specifies the URI which stores the device information externally.

Parent Element: `DevInf`

Restrictions: The content information MUST specify an external device information URI. This element type is optional. If this element occurs in the `DevInf` element, the recipient can retrieve the device information from external storage based on the external URI.

Content Model:

```
<xs:element name="ExtURI" type="xs:anyURI"/>
```

Attributes: None.

Example:

```
<ExtURI>http://www.vendorwebsite.example.com/deviceifo/model.xml</ExtURI>
```

6.2.13 FieldLevel

Usage: Indicates that the sender is able to apply field-level replaces for the corresponding CTType.

Parent Element: `CTCap`

Restrictions: If the sending device has not specified the FieldLevel attribute, or it is specified as "false" in its Device Information for a particular CTType, then the receiving device MUST NOT send field-level changes.

Content Model:

```
<xs:attribute name="FieldLevel" type="xs:boolean" default="false" />
```

Example:

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

6.2.14 FilterCap

Usage: Indicates the filtering capabilities.

Parent Element: `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 `FilterCap` 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:

```
<xs:element name="FilterCap" type="FilterCapType" />

<xs:complexType name="FilterCapType">
  <xs:sequence>
    <xs:element ref="CTType" />
    <xs:element ref="VerCT" />
    <xs:element ref="FilterKeyword" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="PropName" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

Attributes: None.

Example:

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

6.2.15 FilterKeyword

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

Parent Element: `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:

```
<xs:element name="FilterKeyword" type="xs:string" />
```

Attributes: None.

6.2.16 Filter-Rx

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

Parent Element: `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:

```
<xs:element name="Filter-Rx" type="RxTxType" />
```

Attributes: None.

Example:

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

6.2.17 FPUnique

Usage: Specifies if the fingerprints of the data items are unique for one data store.

Parent Element: `SyncCap`

Restrictions: If this attribute is set to "true" (or not present), then it shows that the fingerprints of the data items in the data store are unique. If the fingerprints are unique, then the server can do a more efficient duplicate detection. See [DSPRO] section 5.2.

Content Model:

```
<xs:attribute name="FPUnique" type="xs:boolean" default="true" />
```

Example:

```
<Datastore>
  <SyncCap FPUnique="true">
    ...
  </SyncCap>
  ...
</Datastore>
```

6.2.18 FwV

Usage: Specifies the firmware version of the device.

Parent Element: `DevCap`

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 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.

Content Model:

```
<xs:element name="FwV" type="xs:string" />
```

Attributes: None.

Example:

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

6.2.19 HwV

Usage: Specifies the hardware version of the device.

Parent Element: `DevCap`

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 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.

Content Model:

```
<xs:element name="HwV" type="xs:string" />
```

Attributes: None.

Example:

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

6.2.20 Man

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

Parent Element: `DevCap`

Restrictions: None.

Content Model:

```
<xs:element name="Man" type="xs:string" />
```

Attributes: None.

Example:

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

6.2.21 MaxGUIDSize

Usage: Specifies the maximum size in bytes of a global unique identifier for a given local datastore, which the device is able to receive and store.

Parent Element: `DataStore`

Restrictions: Content information MUST be specified as the 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:

```
<xs:attribute name="MaxGUIDSize" type="xs:unsignedInt" />
```

Attributes: None.

Example: The following is an example of a client capable of receiving and storing a maximum 2 bytes long GUID.

```
<DataStore DisplayName="Addressbook" MaxGUIDSize="32">
  <SourceRef>contacts</SourceRef>
  ...
</DataStore>
```

6.2.22 MaxID

Usage: Specifies the maximum number of items that can be stored in a given local datastore.

Parent Element: `DSMem`

Restrictions: The attribute value is the maximum number of item identifiers that are available for all items in the local datastore.

Content Model:

```
<xs:attribute name="MaxID" type="xs:unsignedInt" />
```

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

```
<DSMem MaxID="1024" />
```

6.2.23 MaxMem

Usage: Specifies the maximum memory size for a given local datastore, in bytes.

Parent Element: `DSMem`

Restrictions: The attribute value is the maximum number of free bytes of memory available in the local database.

Content Model:

```
<xs:attribute name="MaxMem" type="xs:unsignedLong" />
```

Example: The following is an example of 65539 bytes.

```
<DSMem MaxMem="65539"/>
```

6.2.24 MaxOccur

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

Parent Element: `PropInfo`

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:

```
<xs:element name="MaxOccur" type="xs:unsignedInt" />
```

Attributes: None.

Example:

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

6.2.25 MaxSize

Usage: Specifies the maximum size in UTF-8 characters of a given property value.

Parent Element: `PropInfo`

Restrictions: The sender MUST truncate the property to the specified size if the *`Truncate`* attribute is not present, or set to "true". If the *`Truncate`* attribute is set to "false", 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 *`Truncate`* attribute set to "false", a property with an empty value MUST be sent.

Content Model:

```
<xs:element name="MaxSize" type="MaxSizeType" >

<xs:complexType name="MaxSizeType">
  <xs:simpleContent>
    <xs:extension base="xs:unsignedInt">
      <xs:attribute ref="Truncate" use="optional"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

Attributes: *`Truncate`*

Example:

```
<Property>
  <PropName>TEL</PropName>
  <PropInfo>
    <MaxSize>256</MaxSize>
  </PropInfo>
  <PropParam ParamName="TYPE">
    <ValEnum>VOICE,HOME</ValEnum>
    <ValEnum>FAX,HOME</ValEnum>
    <ValEnum>VOICE,CELL</ValEnum>
  </PropParam>
</Property>
```

6.2.26 MaxStoredAnchors

Usage: Specifies the maximum number of sync anchors that might be stored by the device. This can be used to support multiple backup sets, or Undo functionality. The value specified is only a guideline for how many anchors may be stored,

other limits may apply. The supported number of anchors must be at least One, for storing at least the previous (was Next, will be Last) Sync Anchor until the start of the next Sync.

Parent Element: `StoredAnchors`

Restrictions: This attribute SHOULD be present for a device that supports multiple sync anchors.

Content Model:

```
<xs:attribute name="MaxStoredAnchors" type="xs:unsignedInt" />
```

6.2.27 Model

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

Parent Element: `DevCap`

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:

```
<xs:element name="Model" type="xs:string" />
```

Attributes: None.

Example:

```
<Model>1447</Model>
```

6.2.28 OEM

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

Parent Element: `DevInf`

Restrictions: This element type is optional.

Content Model:

```
<xs:element name="OEM" type="xs:string" />
```

Attributes: None.

Example:

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

6.2.29 ParamName

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

Parent Element: `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 attribute MUST be one of the values defined in the table below, or an extension value starting with 'X'. Sending the *`ParamName`* attribute 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:

```
<xs:attribute name="ParamName" type="xs:string" />
```

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"/>
  <PropParam ParamName="ROLE"/>
</Property>
```

6.2.30 Property

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

Parent Element: `CTCap`

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

Content Model:

```
<xs:element name="Property" type="PropertyType" />

<xs:complexType name="PropertyType">
  <xs:sequence>
```

```
    <xs:element ref="PropName"/>
    <xs:element ref="PropInfo" minOccurs="0" />
    <xs:element ref="PropParam" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="ValEnum" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

Attributes: None.

Example:

```
<Property>
  <PropName>TEL</PropName>
  <PropInfo>
    <MaxOccur>3</MaxOccur>
    <MaxSize Truncate="false">255</MaxSize>
  </PropInfo>
  <PropParam ParamName="TYPE">
    <ValEnum>VOICE,HOME</ValEnum>
    <ValEnum>FAX,HOME</ValEnum>
    <ValEnum>VOICE,CELL</ValEnum>
  </PropParam>
</Property>
```

6.2.31 PropInfo

Usage: The placeholder element for some property information elements.

Parent Element: `Property`

Restrictions: This element is optional.

Content Model:

```
<xs:element name="PropInfo" type="PropInfoType" />

<xs:complexType name="PropInfoType">
  <xs:sequence>
    <xs:element ref="DataType" minOccurs="0"/>
    <xs:element ref="MaxOccur" minOccurs="0"/>
    <xs:element ref="MaxSize" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute ref="DisplayName" use="optional"/>
</xs:complexType>
```

Attributes: *`DisplayName`*

Example:

```
<PropInfo>
  <MaxSize Truncate="false">255</MaxSize>
</PropInfo>
```

6.2.32 PropName

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

Parent Element: `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:

```
<xs:element name="PropName" type="xs:string" />
```

Attributes: None.

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

```
<CTCap>
  <CTType>text/x-vcalendar</CTType>
  <VerCT>3.0</VerCT>
  <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>
```

6.2.33 PropParam

Usage: Specifies a supported parameter of a given property

Parent Element: `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:

```
<xs:element name="PropParam" type="PropParamType" />

<xs:complexType name="PropParamType">
  <xs:sequence>
    <xs:element ref="DataType" minOccurs="0"/>
    <xs:element ref="ValEnum" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute ref="DisplayName" use="optional"/>
  <xs:attribute ref="ParamName" use="optional"/>
</xs:complexType>
```

Attributes: *DisplayName*, *ParamName*

Example:

```
<PropParam ParamName="TYPE">
  <ValEnum>VOICE, HOME</ValEnum>
```

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

6.2.34 Rx

Usage: Specifies the supported type and version of a content type received by the specific data store.

Parent Element: `RxTx-CT`

Restrictions: This element type is optional.

Content Model:

```
<xs:element name="Rx" type="RxTxType" />
```

Attributes: None.

Example:

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

6.2.35 Rx-Pref

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

Parent Element: `RxTx-CT`

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

Content Model:

```
<xs:element name="Rx-Pref" type="RxTxType"/>
```

Attributes: None.

Example:

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

6.2.36 RxTx-CT

Usage: The placeholder element for some receiving and transmitting content type information elements.

Parent Element: `DataStore`

Restrictions: This element is mandatory.

Content Model:

```
<xs:element name="RxTx-CT" type="RxTxCTType"/>

<xs:complexType name="RxTxCTType">
  <xs:sequence>
    <xs:element ref="Rx-Pref" />
    <xs:element ref="Rx" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="Tx-Pref" />
    <xs:element ref="Tx" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

Attributes: None.

Example:

```
<RxTx-CT>
  <Rx-Pref> ... <Rx-Pref>
  <Rx> ... <Rx>
  <Tx-Pref> ... <Tx-Pref>
  <Tx> ... <Tx>
</RxTx-CT>
```

6.2.37 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 Element: `DSMem`

Restrictions: When specified as “true” the given datastore memory is shared

Content Model:

```
<xs:attribute name="SharedMem" type="xs:boolean" default="false" />
```

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

```
<DSMem SharedMem="true" MaxMem="65539" MaxID="512" />
```

6.2.38 SourceRef

Usage: Specifies the reference URI for a local data store.

Parent Element: `DataStore`

Restrictions: If the DataStore element type is present, then the SourceRef element type is required. The content information of this element type is the name of the datastore.

Content Model:

```
<xs:element name="SourceRef" type="xs:string" />
```

Attributes: None.

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

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

6.2.39 StoredAnchors

Usage: Top level element for the current and maximum number of stored anchors. Stored anchors are useful for restoring previous data sets.

Parent Element: `SyncCap`

Restrictions: This element SHOULD be present for a device that supports multiple sync anchors, such as a server used for backup and restore.

Content Model:

```
<xs:element name="StoredAnchors" type="StoredAnchorsType"/>

<xs:complexType name="StoredAnchorsType">
  <xs:sequence>
    <xs:element ref="ValidAnchor" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute ref="MaxStoredAnchors"/>
</xs:complexType>
```

Attributes: MaxStoredAnchors

Example: The following is an example of a device declaring that can support up to 5 backup/restore points, with 3 data sets currently available for restore operations.

```
<DevInf xmlns='syncml:devinf'>
  <DevCap>
    ...
  </DevCap>
  <DataStore>
    <SyncCap>
      <StoredAnchors MaxStoredAnchors="5">
        <ValidAnchor>2008-10-19T12:13:00Z</ValidAnchor>
        <ValidAnchor>2008-12-14T15:17:00Z</ValidAnchor>
        <ValidAnchor>2009-02-08T07:19:00Z</ValidAnchor>
      </StoredAnchors>
    </SyncCap>
    ...
  </DataStore>
</DevInf>
```

6.2.40 SupportAtomic

Usage: Specifies that the device supports atomic attribute.

Parent Element: `DevCap`

Restrictions: This attribute MUST be present and specified as "true" for a device that supports the *`atomic`* attribute.

Content Model:

```
<xs:attribute name="SupportAtomic" type="xs:boolean" default="false" />
```

6.2.41 SupportEncryption

Usage: Specifies that the device supports data encryption.

Parent Element: `DevCap`

Restrictions: This attribute MUST be present and specified as "true" for a device that supports data encryption.

Content Model:

```
<xs:attribute name="SupportEncryption" type="xs:boolean" default="false"/>
```

6.2.42 SupportFieldLevel

Usage: Specifies that the device supports field level replace.

Parent Element: `DevCap`

Restrictions: This attribute MUST be present and specified as "true" for a device that supports field level replace.

Content Model:

```
<xs:attribute name="SupportFieldLevel" type="xs:boolean" default="false"/>
```

6.2.43 SupportHierarchicalSync

Usage: Indicates the support for hierarchical sync. Note that this does not indicate the type of hierarchical contents that are supported.

Parent Element: `SyncCap`

Restrictions: This attribute MUST be present and specified as "true" for a Datastore that supports hierarchical synchronisation.

Content Model:

```
<xs:attribute name="SupportHierarchicalSync"
  type="xs:boolean" default="false" />
```

Example:

```
<DevInf xmlns='syncml:devinf' Version="2.0">
  <DevCap>
    ...
  </DevCap>
  <DataStore>
      ...
    <SyncCap SupportHierarchicalSync="true">
      ...
    </SyncCap>
```

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

6.2.44 SupportLargeObjs

Usage: Specifies that the device supports receiving large objects.

Parent Element: `DevCap`

Restrictions: This attribute MUST be absent, or present and specified as "true" for a device that supports large objects. This attribute MUST be present and specified as "false" for a device that does not support large objects.

Content Model:

```
<xs:attribute name="SupportLargeObjs" type="xs:boolean" default="true" />
```

Example:

```
<DevInf xmlns='syncml:devinf' Version="2.0">
  <DevCap SupportLargeObjs="true">
    <Man>SomeManufacturer</Man>
    <Model>SomeModel</Model>
    <OEM>MyOEM</OEM>
    <FwV>1.0</FwV>
    <SwV>2.99</SwV>
    <HwV>2.1</HwV>
    <DevID>1218182THD012345-2</DevID>
    <DevType>pager</DevType>
  </DevCap>
  <DataStore>
  ...
  </DataStore>
</DevInf>
```

6.2.45 SupportNumberOfChanges

Usage: Specifies that the device supports `NumberOfChanges` Element (see [DSSYNTAX]).

Parent Element: `DevCap`

Restrictions:

If the client does not specify, or specifies *`SupportNumberOfChanges`* as "true" in its Device Information, then the server MUST send the `NumberOfChanges` element (see section "NumberOfChanges" in [DSSYNTAX]).

If the client specifies *`SupportNumberOfChanges`* as "false" in its Device Information, then the server MUST NOT send the `NumberOfChanges` element.

Content Model:

```
<xs:attribute name="SupportNumberOfChanges"
  type="xs:boolean" default="true"/>
```

Example:

```
<DevInf xmlns='syncml:devinf' Version="2.0">
  <DevCap SupportNumberOfChanges="true" >
    <Man>SomeManufacturer</Man>
    <Model>SomeModel</Model>
    <OEM>MyOEM</OEM>
    <FwV>1.0</FwV>
    <SwV>2.99</SwV>
    <HwV>2.1</HwV>
    <DevID>1218182THD012345-2</DevID>
    <DevType>pager</DevType>
  </DevCap>
  <DataStore>
   ...
  </DataStore>
</DevInf>
```

6.2.46 SupportSequence

Usage: Specifies that the device supports the *sequence* attribute.

Parent Element: DevCap

Restrictions: This attribute MUST be present and specified as "true" for a device that supports the *sequence* attribute.

Content Model:

```
<xs:attribute name="SupportSequence" type="xs:boolean" default="false" />
```

6.2.47 SupportSftDel

Usage: Specifies that the device supports soft deletion.

Parent Element: DevCap

Restrictions: This attribute MUST be present and specified as "true" for a device that supports soft deletion.

Content Model:

```
<xs:attribute name="SupportSftDel" type="xs:boolean" default="false" />
```

6.2.48 SwV

Usage: Specifies the software version of the device.

Parent Element: DevCap

Restrictions:

This element type is mandatory, and must uniquely identify the specific software build or version.

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:

```
<xs:element name="SwV" type="xs:string" />
```

Attributes: None.

Example:

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

6.2.49 SyncCap

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

Parent Element: `DataStore`

Restrictions: This element is mandatory.

Content Model:

```
<xs:element name="SyncCap" type="SyncCapType"/>

<xs:complexType name="SyncCapType">
  <xs:sequence>
    <xs:element ref="StoredAnchors" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute ref="FPUnique" use="optional"/>
  <xs:attribute ref="SupportHierarchicalSync" use="optional"/>
</xs:complexType>
```

Attributes: *FPUnique*, *SupportHierarchicalSync*

Example: The following is an example declaring that the fingerprints are unique for a particular datastore.

```
<DataStore>
  ...
  <SyncCap FPUnique="true">
    ...
  </SyncCap>
</DataStore>
```

6.2.50 Truncate

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 with Truncate set to false, the property MUST NOT be sent.

Parent Element: `MaxSize`

Restrictions: If the sending device has specified the Truncate attribute with a value of false 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:

```
<xs:attribute name="Truncate" type="xs:boolean" default="true"/>
```

Attributes: None.

Example:

```
<Property>
  <PropName>TEL</PropName>
  <PropInfo>
    <MaxSize Truncate="false">255</MaxSize>
  </PropInfo>
  <PropParam ParamName="TYPE">
    <ValEnum>VOICE,HOME</ValEnum>
    <ValEnum>FAX,HOME</ValEnum>
    <ValEnum>VOICE,CELL</ValEnum>
  </PropParam>
</Property>
```

6.2.51 Tx

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

Parent Element: `RxTx-CT`

Restrictions: This element type is optional.

Content Model:

```
<xs:element name="Tx" type="RxTxType" />
```

Attributes: None.

Example:

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

6.2.52 Tx-Pref

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

Parent Element: `RxTx-CT`

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

Content Model:

```
<xs:element name="Tx-Pref" type="RxTxType"/>
```

Attributes: None.

Example:

```
<Tx-Pref>
```

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

6.2.53 UTC

Usage: Specifies that the device supports UTC based time.

Parent Element: `DevCap`

Restrictions: If UTC attribute is present and specified as "true", 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:

```
 <xs:attribute name="UTC" type="xs:boolean" default="false" />
```

Example:

```
<DevInf xmlns='syncml:devinf' "Version=2.0">
  <DevCap UTC="true">
    ...
  </DevCap>
  <DataStore>
   ...
  </DataStore>
</DevInf>
```

6.2.54 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:

```
<xs:element name="ValEnum" type="xs:string" />
```

Attributes: None.

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

```
<CTCap>
  <CTType>text/calendar</CTType>
  <VerCT>2.0</VerCT>
  <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>
```

6.2.55 ValidAnchor

Usage: Specifies a currently valid Sync Anchor. This may be used as the Next Sync Anchor of a one way sync, to specify loading data from that point in time (Such as for Undo Functionality), or may be used as a valid Last Sync Anchor.

Parent Element: `StoredAnchors`

Restrictions: This element SHOULD be present for a device that supports multiple sync anchors, for each valid stored sync anchor.

Content Model:

```
<xs:element name="ValidAnchor" type="AnchorType " />
<xs:simpleType name="AnchorType">
  <xs:union memberTypes="xs:dateTime xs:positiveInteger"/>
</xs:simpleType>
```

Attributes: None.

6.2.56 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:

```
 <xs:element name="VerCT" type="xs:string" />
```

Attributes: None.

Example:

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

6.2.57 Version

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

Parent Element: `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 2.0. The attribute MUST be included in the `DevInf` element.

Content Model:

```
<xs:attribute name="Version" type="xs:string" />
```

Example:

```
<DevInf xmlns='syncml:devinf' Version="2.0">
  <DevCap>
    ...
  </DevCap>
  ...
</DevInf>
```

6.2.58 XName

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

Parent Element: `Ext`

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

Content Model:

```
 <xs:element name="XName" type="xs:string" />
```

Attributes: None.

Example:

```
<Ext>
  <XName>X-Foo-CliVer</XName>
  <XValue>5.0</XValue>
  <XValue>5.01</XValue>
  <XValue>5.02</XValue>
</Ext>
```

6.2.59 XValue

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

Parent Element: `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:

```
<xs:element name="XValue" type="xs:string" />
```

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>
  <XName>X-Bar-Enum</XName>
```

```
  <XValue>1</XValue>
  <XValue>..</XValue>
  <XValue>5</XValue>
</Ext>
```

7. Device Information Schema

Please refer to the DevInf Schema specification.

8. WBXML Definitions

This version of the DevInf Schema specification is associated with the WBXML code space 0x1206 and the formal public identifier -//SyncML//Schema DevInf 2.0//EN.

8.1 Elements

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

Element Type Name	WBXML Tag Token (Hex Value)
CTCap	05
CTType	06
DataStore	07
DataType	08
DevID	09
DevInf	0A
DevType	0B
Reserved for future use	0C
DSMem	0D
Ext	0E
FwV	0F
HwV	10
Man	11
Reserved for future use	12 ~ 14
Model	15
OEM	16
Reserved for future use	17
PropName	18
Rx	19
Rx-Pref	1A
Reserved for future use	1B
MaxSize	1C
SourceRef	1D
SwV	1E
SyncCap	1F
ExtURI	20

Tx	21
Tx-Pref	22
ValEnum	23
VerCT	24
Reserved for future use	25
XName	26
XValue	27
Reserved for future use	28 ~ 2A
Property	2B
PropParam	2C
MaxOccur	2D
Reserved for future use	2E ~ 2F
Filter-Rx	30
FilterCap	31
FilterKeyword	32
Reserved for future use	33 ~ 34
DevCap	35
PropInfo	36
RxTx-CT	37
StoredAnchors	38
ValidAnchor	39
Reserved for future use	3A ~ 3F

Table 1: WBXML Element Token Definitions – Tag Order

Element Type Name	WBXML Tag Token (Hex Value)
CTCap	05
CTType	06
DataStore	07
DataType	08
DevCap	35
DevID	09
DevInf	0A
DevType	0B
DSMem	0D
Ext	0E
ExtURI	20
FilterCap	31
FilterKeyword	32
Filter-Rx	30
FwV	0F
HwV	10
Man	11
MaxOccur	2D
MaxSize	1C
Model	15
OEM	16
Property	2B
PropInfo	36
PropName	18
PropParam	2C
Reserved for future use	0C
Reserved for future use	12 ~ 14
Reserved for future use	17
Reserved for future use	1B
Reserved for future use	25
Reserved for future use	28 ~ 2A
Reserved for future use	2E ~ 2F

Element Type Name	WBXML Tag Token (Hex Value)
Reserved for future use	33 ~ 34
Reserved for future use	3A ~ 3F
Rx	19
Rx-Pref	1A
RxTx-CT	37
SourceRef	1D
StoredAnchors	38
SwV	1E
SyncCap	1F
Tx	21
Tx-Pref	22
ValEnum	23
ValidAnchor	39
VerCT	24
XName	26
XValue	27

Table 2: WBXML Element Token Definitions – Alphabetical Order

8.2 Attribute Start Tokens

The following WBXML token codes represent the start of an attribute (some including value) in code page 0x00, DevInf Schema.

Attribute Name	Attribute Value Prefix	WBXML Tag Token (Hex Value)
DisplayName		05
FieldLevel	false	06
FieldLevel	true	07
FPUnique	false	08
FPUnique	true	09
MaxGUIDSize		0A
MaxID		0B
MaxMem		0C
MaxStoredAnchors		0D
ParamName		0E
SharedMem	false	0F
SharedMem	true	10
SupportAtomic	false	11
SupportAtomic	true	12
SupportEncryption	false	13
SupportEncryption	true	14
SupportFieldLevel	false	15
SupportFieldLevel	true	16
SupportHierarchicalSync	false	17
SupportHierarchicalSync	true	18
SupportLargeObjs	false	19
SupportLargeObjs	true	1A
SupportNumberOfChanges	false	1B
SupportNumberOfChanges	true	1C
SupportSequence	false	1D
SupportSequence	true	1E
SupportSftDel	false	1F

Attribute Name	Attribute Value Prefix	WBXML Tag Token (Hex Value)
SupportSftDel	true	20
Truncate	false	21
Truncate	true	22
UTC	false	23
UTC	true	24
Version		25
Reserved for future use		26 ~ 3F
Reserved for future use		45 ~ 7F

Table 3: Attribute Start Token Definitions – Alphabetical

8.3 Attribute Value Tokens

The following WBXML token codes represent attribute values in code page 0x00, DevInf Schema.

Element Type Name	WBXML Tag Token (Hex Value)
A	85
ATTACHTYPE	86
CN	87
CUTYPE	88
D	89
DELEGATED-FROM	8A
DELEGATED-TO	8B
DIR	8C
ENC	8D
EXPECT	8E
H	8F
LANGUAGE	90
MEMBER	91
PARTSTAT	92
R	93
ROLE	94
RSVP	95
S	96

Element Type Name	WBXML Tag Token (Hex Value)
SENT-BY	97
STATUS	98
TEXTTYPE	99
TYPE	9A
W	9B
X	9C
Reserved for future use	9D ~ BF
Reserved for future use	C5 ~ FF

Table 4: WBXML Attribute Value Token Definitions

9. EXAMPLES

9.1 XML

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 as specified in section 5.1 on the `DevInf` element type.

```
<DevInf xmlns='syncml:devinf' Version="2.0">
  <DevCap SupportLargeObjs="true"
    SupportNumberOfChanges="true" UTC="true">
    <Man>Big Factory, Ltd.</Man>
    <Model>4119</Model>
    <OEM>Jane's phones</OEM>
    <FwV>2.0e</FwV>
    <SwV>2.0</SwV>
    <HwV>1.22I</HwV>
    <DevID>1218182THD000001-2</DevID>
    <DevType>phone</DevType>
  </DevCap>
  <DataStore DisplayName="Phonebook" MaxGUIDSize="32">
    <SourceRef>contacts</SourceRef>
    <RxTx-CT>
      <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>
    </RxTx-CT>
    <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">
          <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">
          <ValEnum>VOICE,HOME</ValEnum>
          <ValEnum>FAX,HOME</ValEnum>
          <ValEnum>VOICE,CELL</ValEnum>
        </PropParam>
      </Property>
    </CTCap>
    <SyncCap FPUnique="true"/>
    <DSMem MaxMem="32650" MaxID="250" />
  </DataStore>
  <Ext>
    <XName>startmsg</XName>
    <XValue>Hello World</XValue>
  </Ext>
  <Ext>
    <XName>endmsg</XName>
    <XValue>Goodbye</XValue>
  </Ext>
</DevInf>
```

9.2 WBXML

The following is a complete message of an HTTP Post including the WBXML representation for the device information object specified in section 9.1. This example uses opaque data and inline strings. The example also assumes that the character encoding is UTF-8.

```
0000  00 40 05 32 06 16 00 16 b6 89 63 a4 08 00 45 80   .@.2......c...E.
0010  03 f8 d1 8a 40 00 6c 06 b5 12 4b df 76 29 c0 a8   ....@.l...K.v)..
0020  01 32 0e bd 00 50 63 3e ef 26 27 b1 4d cf 50 18   .2...Pc>.&'.M.P.
0030  80 c4 00 e2 00 00 50 4f 53 54 20 2f 53 79 6e 63   ......POST /Sync
0040  4d 4c 20 48 54 54 50 2f 31 2e 31 0d 0a 41 63 63   ML HTTP/1.1..Acc
0050  65 70 74 2d 4c 61 6e 67 75 61 67 65 3a 20 65 6e   ept-Language: en
0060  2d 75 73 0d 0a 43 6f 6e 74 65 6e 74 2d 54 79 70   -us..Content-Typ
0070  65 3a 20 61 70 70 6c 69 63 61 74 69 6f 6e 2f 76   e: application/v
0080  6e 64 2e 73 79 6e 63 6d 6c 2d 64 65 76 69 6e 66   nd.syncml-devinf
0090  2b 77 62 78 6d 6c 0d 0a 55 73 65 72 2d 41 67 65   +wbxml..User-Age
00a0  6e 74 3a 20 44 65 76 49 6e 66 2d 45 78 61 6d 70   nt: DevInf-Examp
00b0  6c 65 0d 0a 48 6f 73 74 3a 20 64 63 68 61 6d 70   le..Host: dchamp
00c0  61 67 6e 65 2e 63 6f 6d 0d 0a 43 6f 6e 74 65 6e   agne.com..Conten
00d0  74 2d 4c 65 6e 67 74 68 3a 20 36 30 39 0d 0a 43   t-Length: 609..C
00e0  6f 6e 6e 65 63 74 69 6f 6e 3a 20 4b 65 65 70 2d   onnection: Keep-
00f0  41 6c 69 76 65 0d 0a 43 61 63 68 65 2d 43 6f 6e   Alive..Cache-Con
0100  74 72 6f 6c 3a 20 6e 6f 2d 63 61 63 68 65 0d 0a   trol: no-cache..
0110  0d 0a 03 a4 06 6a 00 ca 25 03 32 2e 30 00 01 f5   .....j..%.2.0...
0120  1a 1c 24 01 51 c3 12 42 69 67 20 46 61 63 74 6f   ..$.Q..Big Facto
0130  72 79 2c 20 4c 74 64 2e 00 01 55 03 34 31 31 39   ry, Ltd...U.4119
0140  00 01 56 03 4a 61 6e 65 27 73 20 70 68 6f 6e 65   ..V.Jane's phone
0150  73 00 01 4f 03 32 2e 30 65 00 01 5e 03 32 2e 30   s..O.2.0e..^.2.0
0160  00 01 50 03 31 2e 32 32 49 00 01 49 03 31 32 31   ..P.1.22I..I.121
0170  38 31 38 32 54 48 44 30 30 30 30 30 31 2d 32 00   8182THD000001-2.
0180  01 4b 03 70 68 6f 6e 65 00 01 01 c7 05 03 50 68   .K.phone......Ph
0190  6f 6e 65 62 6f 6f 6b 00 0a 03 33 32 00 01 5d 03   onebook...32..].
01a0  63 6f 6e 74 61 63 74 73 00 01 77 5a 46 03 74 65   contacts..wZF.te
01b0  78 74 2f 76 63 61 72 64 00 01 64 03 33 2e 30 00   xt/vcard..d.3.0.
01c0  01 01 59 46 03 74 65 78 74 2f 78 2d 76 63 61 72   ..YF.text/x-vcar
01d0  64 00 01 64 03 32 2e 31 00 01 01 62 46 03 74 65   d..d.2.1...bF.te
01e0  78 74 2f 76 63 61 72 64 00 01 64 03 33 2e 30 00   xt/vcard..d.3.0.
01f0  01 01 61 46 03 74 65 78 74 2f 78 2d 76 63 61 72   ..aF.text/x-vcar
0200  64 00 01 64 03 32 2e 31 00 01 01 01 45 46 03 74   d..d.2.1....EF.t
0210  65 78 74 2f 76 63 61 72 64 00 01 64 03 33 2e 30   ext/vcard..d.3.0
0220  00 01 6b 58 03 42 45 47 49 4e 00 01 63 03 56 43   ..kX.BEGIN..c.VC
0230  41 52 44 00 01 01 6b 58 03 45 4e 44 00 01 63 03   ARD...kX.END..c.
0240  56 43 41 52 44 00 01 01 6b 58 03 56 45 52 53 49   VCARD...kX.VERSI
0250  4f 4e 00 01 63 03 33 2e 30 00 01 01 6b 58 03 4e   ON..c.3.0...kX.N
0260  00 01 01 6b 58 03 54 45 4c 00 01 ec 0e 9a 01 63   ...kX.TEL......c
0270  03 56 4f 49 43 45 2c 48 4f 4d 45 00 01 63 03 46   .VOICE,HOME..c.F
0280  41 58 2c 48 4f 4d 45 00 01 63 03 56 4f 49 43 45   AX,HOME..c.VOICE
0290  2c 43 45 4c 4c 00 01 01 01 01 45 46 03 74 65 78   ,CELL.....EF.tex
02a0  74 2f 78 2d 76 63 61 72 64 00 01 64 03 32 2e 31   t/x-vcard..d.2.1
02b0  00 01 6b 58 03 42 45 47 49 4e 00 01 63 03 56 43   ..kX.BEGIN..c.VC
02c0  41 52 44 00 01 01 6b 58 03 45 4e 44 00 01 63 03   ARD...kX.END..c.
02d0  56 43 41 52 44 00 01 01 6b 58 03 56 45 52 53 49   VCARD...kX.VERSI
02e0  4f 4e 00 01 63 03 32 2e 31 00 01 01 6b 58 03 4e   ON..c.2.1...kX.N
02f0  00 01 01 6b 58 03 54 45 4c 00 01 ec 0e 9a 01 63   ...kX.TEL......c
0300  03 56 4f 49 43 45 2c 48 4f 4d 45 00 01 63 03 46   .VOICE,HOME..c.F
0310  41 58 2c 48 4f 4d 45 00 01 63 03 56 4f 49 43 45   AX,HOME..c.VOICE
```

```
0320  2c 43 45 4c 4c 00 01 01 01 01 9f 09 01 8d 0c 03   ,CELL...........
0330  33 32 36 35 30 00 0b 03 32 35 30 00 01 01 4e 66   32650...250...Nf
0340  03 73 74 61 72 74 6d 73 67 00 01 67 03 48 65 6c   .startmsg..g.Hel
0350  6c 6f 20 57 6f 72 6c 64 00 01 01 4e 66 03 65 6e   lo World...Nf.en
0360  64 6d 73 67 00 01 67 03 47 6f 6f 64 62 79 65 00   dmsg..g.Goodbye.
0370  01 01 01                                          ...
```

In an expanded and annotated form:

```
Frame 1 (883 bytes on wire, 883 bytes captured)
Ethernet II, ... (Bytes 0x00-0x0D above)

Internet Protocol ... (Bytes 0x0E-0x21 above)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x80 (DSCP 0x20: Class Selector 4; ECN: 0x00)
    Total Length: 1016
    Identification: 0xd18a (53642)
    Flags: 0x04 (Don't Fragment)
    Fragment offset: 0
    Time to live: 108
    Protocol: TCP (0x06)
    Header checksum: 0xb512 [correct]

Transmission Control Protocol, (Bytes 0x22-0x35 above)
    Src Port: 3773, Dst Port: http (80), Seq: 1, Ack: 1, Len: 829
    Header length: 20 bytes
    Flags: 0x18 (PSH, ACK)
    Window size: 32964
    Checksum: 0xe4fd [correct]

Hypertext Transfer Protocol (Bytes 0x36-0x111 above)
    POST /SyncML HTTP/1.1\r\n
    Accept-Language: en-us\r\n
    Content-Type: application/vnd.syncml-devinf+wbxml\r\n
    User-Agent: DevInf-Example\r\n
    Host: dchampagne.com\r\n
    Content-Length: 609\r\n
    Connection: Keep-Alive\r\n
    Cache-Control: no-cache\r\n
    \r\n

WAP Binary XML, (Bytes 0x112-0x372 Above)
    Version: 1.3 (0x03)
    Public Identifier (known): -//SYNCML//Schema DevInf 2.0//EN (0x00001206)
    Character Set: utf-8 (0x0000006a)
    String table: 0 bytes
```

Token Stream	Description
03	Version number – WBXML v1.3
A4 06	FPI for 0x1206, mb_u_int32 --> 0000 - not needed 0001 0010 --> 010 0100 | 0x80 0000 0110 --> 000 0110 & 0x7F
6A	Charset is UTF-8

00	String table length
	No String table (length 0)
CA	<DevInf> Attributes and Content 0x0A | 0x80 | 0x40
	No Namespace, already have FPI
25	Version=
03	Inline string follows
'2' '.' '0'	String "2.0"
00	Null Terminator of String
01	END of DevInf attributes
F5	<DevCap> Attributes and Content
1A	SupportLargeObjs="true"
1C	SupportNumberOfChanges="true"
24	UTC="true"
01	END of DevCap attributes
51	<Man> Content follows 0x11 | 0x40
C3	Opaque data follows
12	Length of opaque data
'B' 'i' 'g' ' ' 'F' 'a' 'c' 't' 'o' 'r' 'y' ',' ' ' 'L' 't' 'd' '.'	String "Big Factory, Ltd."
00	Null Terminator of String (Not required with Opaque Data)
01	</Man>
55	<Model> Content follows
03	Inline string follows
'4' '1' '1' '9'	String "4119"
00	Null Terminator of String
01	</Model>
56	<OEM> Content follows
03	Inline string follows
'J' 'a' 'n' 'e' '\'' 's' ' ' 'p' 'h' 'o' 'n' 'e' 's'	String "Jane's phones"
00	Null Terminator of String
01	</OEM>
4F	<FwV> Content follows
03	Inline string follows
'2' '.' '0' 'e'	String "2.0e"
00	Null Terminator of String
01	</FwV>
5E	<SwV> Content follows
03	Inline string follows
'2' '.' '0'	String "2.0"
00	Null Terminator of String
01	</SwV>
50	<HwV> Content follows
03	Inline string follows
'1' '.' '2' '2' 'I'	String "1.22I"
00	Null Terminator of String
01	</HwV>
49	<DevID> Content follows
03	Inline string follows
'1' '2' '1' '8' '1' '8' '2' 'T' 'H' 'D' '0' '0' '0' '0' '1' '-' '2'	String "1218182THD000001-2"
00	Null Terminator of String
01	</DevID>

4B	<DevType> Content follows
03	Inline string follows
'p' 'h' 'o' 'n' 'e'	String "phone"
00	Null Terminator of String
01	</DevType>
01	</DevCap>
C7	<DataStore> Attributes and Content
05	DisplayName=
03	Inline string follows
'P' 'h' 'o' 'n' 'e' 'b' 'o' 'o' 'k'	String "Phonebook"
00	Null Terminator of String
0A	MaxGUIDSize=
03	Inline string follows
'3' '2'	String "32"
00	Null Terminator of String
01	END of DataStore attributes
5D	<SourceRef> Content follows
03	Inline string follows
'c' 'o' 'n' 't' 'a' 'c' 't' 's'	String "contacts"
00	Null Terminator of String
01	</SourceRef>
77	<RxTx-CT> Content follows
5A	<Rx-Pref> Content follows
46	<CTType> Content follows
03	Inline string follows
't' 'e' 'x' 't' '\/' 'v' 'c' 'a' 'r' 'd'	String "text/vcard"
00	Null Terminator of String
01	</CTType>
64	<VerCT> Content follows
03	Inline string follows
'3' '.' '0'	String "3.0"
00	Null Terminator of String
01	</VerCT>
01	</Rx-Pref>
59	<Rx> Content follows
46	<CTType> Content follows
03	Inline string follows
't' 'e' 'x' 't' '\/' 'x' '-' 'v' 'c' 'a' 'r' 'd'	String "text/x-vcard"
00	Null Terminator of String
01	</CTType>
64	<VerCT> Content follows
03	Inline string follows
'2' '.' '1'	String "2.1"
00	Null Terminator of String
01	</VerCT>
01	</Rx>
62	<Tx-Pref> Content follows
46	<CTType> Content follows
03	Inline string follows
't' 'e' 'x' 't' '\/' 'v' 'c' 'a' 'r' 'd'	String "text/vcard"
00	Null Terminator of String
01	</CTType>
64	<VerCT> Content follows
03	Inline string follows

'3' '.' '0'	String "3.0"
00	Null Terminator of String
01	</VerCT>
01	</Tx-Pref>
61	<Tx> Content follows
46	<CTType> Content follows
03	Inline string follows
't' 'e' 'x' 't' '\/' 'x' '-' 'v' 'c' 'a' 'r' 'd'	String "text/x-vcard"
00	Null Terminator of String
01	</CTType>
64	<VerCT> Content follows
03	Inline string follows
'2' '.' '1'	String "2.1"
00	Null Terminator of String
01	</VerCT>
01	</Tx>
01	</RxTx-CT>
45	<CTCap> Content follows
46	<CTType> Content follows
03	Inline string follows
't' 'e' 'x' 't' '\/' 'v' 'c' 'a' 'r' 'd'	String "text/vcard"
00	Null Terminator of String
01	</CTType>
64	<VerCT> Content follows
03	Inline string follows
'3' '.' '0'	String "3.0"
00	Null Terminator of String
01	</VerCT>
6B	<Property> Content follows
58	<PropName> Content follows
03	Inline string follows
'B' 'E' 'G' 'I' 'N'	String "BEGIN"
00	Null Terminator of String
01	</PropName>
63	<ValEnum>
03	Inline string follows
'V' 'C' 'A' 'R' 'D'	String "VCARD"
00	Null Terminator of String
01	</ValEnum>
01	</Property>
6B	<Property> Content follows
58	<PropName> Content follows
03	Inline string follows
'E' 'N' 'D'	String "END"
00	Null Terminator of String
01	</PropName>
63	<ValEnum>
03	Inline string follows
'V' 'C' 'A' 'R' 'D'	String "VCARD"
00	Null Terminator of String
01	</ValEnum>
01	</Property>
6B	<Property> Content follows
58	<PropName> Content follows

03	Inline string follows
'V' 'E' 'R' 'S' 'I' 'O' 'N'	String "VERSION"
00	Null Terminator of String
01	</PropName>
63	<ValEnum>
03	Inline string follows
'3' '.' '0'	String "3.0"
00	Null Terminator of String
01	</ValEnum>
01	</Property>
6B	<Property> Content follows
58	<PropName> Content follows
03	Inline string follows
'N'	String "N"
00	Null Terminator of String
01	</PropName>
01	</Property>
6B	<Property> Content follows
58	<PropName> Content follows
03	Inline string follows
'T' 'E' 'L'	String "TEL"
00	Null Terminator of String
01	</PropName>
EC	<PropParam> Attributes and Content
0E	ParamName=
9A	Attribute Value "TYPE"
01	END of PropParam attributes
63	<ValEnum>
03	Inline string follows
'V' 'O' 'I' 'C' 'E' ',' 'H' 'O' 'M' 'E'	String "VOICE,HOME"
00	Null Terminator of String
01	</ValEnum>
63	<ValEnum>
03	Inline string follows
'F' 'A' 'X' ',' 'H' 'O' 'M' 'E'	String "FAX,HOME"
00	Null Terminator of String
01	</ValEnum>
63	<ValEnum>
03	Inline string follows
'V' 'O' 'I' 'C' 'E' ',' 'C' 'E' 'L' 'L'	String "VOICE,CELL"
00	Null Terminator of String
01	</ValEnum>
01	</PropParam>
01	</Property>
01	</CTCap>
45	<CTCap> Content follows
46	<CTType> Content follows
03	Inline string follows
't' 'e' 'x' 't' '\/' 'x' '-' 'v' 'c' 'a' 'r' 'd'	String "text/x-vcard"
00	Null Terminator of String
01	</CTType>
64	<VerCT> Content follows
03	Inline string follows
'2' '.' '1'	String "2.1"

00	Null Terminator of String
01	</VerCT>
6B	<Property> Content follows
58	<PropName> Content follows
03	Inline string follows
'B' 'E' 'G' 'I' 'N'	String "BEGIN"
00	Null Terminator of String
01	</PropName>
63	<ValEnum>
03	Inline string follows
'V' 'C' 'A' 'R' 'D'	String "VCARD"
00	Null Terminator of String
01	</ValEnum>
01	</Property>
6B	<Property> Content follows
58	<PropName> Content follows
03	Inline string follows
'E' 'N' 'D'	String "END"
00	Null Terminator of String
01	</PropName>
63	<ValEnum>
03	Inline string follows
'V' 'C' 'A' 'R' 'D'	String "VCARD"
00	Null Terminator of String
01	</ValEnum>
01	</Property>
6B	<Property> Content follows
58	<PropName> Content follows
03	Inline string follows
'V' 'E' 'R' 'S' 'I' 'O' 'N'	String "VERSION"
00	Null Terminator of String
01	</PropName>
63	<ValEnum>
03	Inline string follows
'2' '.' '1'	String "2.1"
00	Null Terminator of String
01	</ValEnum>
01	</Property>
6B	<Property> Content follows
58	<PropName> Content follows
03	Inline string follows
'N'	String "N"
00	Null Terminator of String
01	</PropName>
01	</Property>
6B	<Property> Content follows
58	<PropName> Content follows
03	Inline string follows
'T' 'E' 'L'	String "TEL"
00	Null Terminator of String
01	</PropName>
EC	<PropParam> Attributes and Content
0E	ParamName=
9A	Attribute Value "TYPE"

01	END of PropParam attributes
63	<ValEnum>
03	Inline string follows
'V' 'O' 'I' 'C' 'E' ',' 'H' 'O' 'M' 'E'	String "VOICE,HOME"
00	Null Terminator of String
01	</ValEnum>
63	<ValEnum>
03	Inline string follows
'F' 'A' 'X' ',' 'H' 'O' 'M' 'E'	String "FAX,HOME"
00	Null Terminator of String
01	</ValEnum>
63	<ValEnum>
03	Inline string follows
'V' 'O' 'I' 'C' 'E' ',' 'C' 'E' 'L' 'L'	String "VOICE,CELL"
00	Null Terminator of String
01	</ValEnum>
01	</PropParam>
01	</Property>
01	</CTCap>
9F	<SyncCap> Attributes follow
09	FPUnique="true"
01	END of SyncCap (tag + attributes)
8D	<DSMem> Attributes follow
0C	MaxMem=
03	Inline string follows
'3' '2' '6' '5' '0'	String "32650"
00	Null Terminator of String
0B	MaxID=
03	Inline string follows
'2' '5' '0'	String "250"
00	Null Terminator of String
01	END of DSMem (tag + attributes)
01	</DataStore>
4E	<Ext> Content follows
66	<XName> Content follows
03	Inline string follows
's' 't' 'a' 'r' 't' 'm' 's' 'g'	String "startmsg"
00	Null Terminator of String
01	</XName>
67	<XValue> Content follows
03	Inline string follows
'H' 'e' 'l' 'l' 'o' ' ' 'W' 'o' 'r' 'l' 'd'	String "Hello World"
00	Null Terminator of String
01	</XValue>
01	</Ext>
4E	<Ext> Content follows
66	<XName> Content follows
03	Inline string follows
'e' 'n' 'd' 'm' 's' 'g'	String "endmsg"
00	Null Terminator of String
01	</XName>
67	<XValue> Content follows
03	Inline string follows
'G' 'o' 'o' 'd' 'b' 'y' 'e'	String "Goodbye"
00	Null Terminator of String

01	</XValue>
01	</Ext>
01	</DevInf>

Note some valid alternatives for the beginning of the WBXML document:

Token Stream	Description
02	Version number – WBXML v1.2
00	FPI for DTD in string table
00	index into string table for the identifier
6A	Charset is UTF-8
21	String table length
'-' '/' '/' 'S' 'y' 'n' 'c' 'M' 'L' '/' '/' 'S' 'c' 'h' 'e' 'm' 'a' ' ' 'D' 'e' 'f' 'I' 'n' 'f' ' ' '2' '.' '0' '/' '/' 'E' 'N' 0x00	-//SyncML//Schema DevInf 2.0//EN

Note also that Null terminated strings may be specified either as inline strings, or as opaque data. Data that may include 0x00 MUST be specified as opaque data. Additionally, note that additional strings may be encoded into the string table, and included by reference throughout the document. However, since few parsers used in earlier versions of OMA-DS supported this, this use is not recommended.

10.MIME Media Type Registration

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

10.1 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, verschema

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.

verschema 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 "VerSchema" 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.

verschema-param = ";" "verschema" "=" 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

10.2 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, verschema

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.

verschema 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 "VerSchema" 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.

verschema-param = ";" "verschema" "=" 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	6.2.1	O	
SCR-DS-DEVINF-C-002	Support for CTType element	6.2.2	M	
SCR-DS-DEVINF-C-003	Support for DataStore element	6.2.3	M	
SCR-DS-DEVINF-C-004	Support for DataType element	6.2.4	O	
SCR-DS-DEVINF-C-005	Support for DevCap element	6.2.5	M	
SCR-DS-DEVINF-C-006	Support for DevID element	6.2.6	M	
SCR-DS-DEVINF-C-007	Support for DevInf element	6.2.7	M	
SCR-DS-DEVINF-C-008	Support for DevType element	6.2.8	M	
SCR-DS-DEVINF-C-009	Support for DisplayName attribute	6.2.9	O	
SCR-DS-DEVINF-C-010	Support for DSMem element	6.2.10	O	
SCR-DS-DEVINF-C-011	Support for Ext element	6.2.11	O	
SCR-DS-DEVINF-C-012	Support for ExtURI element	6.2.12	O	
SCR-DS-DEVINF-C-013	Support for FieldLevel attribute	6.2.13	O	
SCR-DS-DEVINF-C-014	Support for Filter-Rx element	6.2.16	O	
SCR-DS-DEVINF-C-015	Support for FilterCap element	6.2.14	O	
SCR-DS-DEVINF-C-016	Support for FPUnique attribute	6.2.17	O	
SCR-DS-DEVINF-C-017	Support for FilterKeyword element	6.2.15	O	
SCR-DS-DEVINF-C-018	Support for FwV element	6.2.18	M	
SCR-DS-DEVINF-C-019	Support for HwV element	6.2.19	M	
SCR-DS-DEVINF-C-020	Support for Man element	6.2.20	M	
SCR-DS-DEVINF-C-021	Support for MaxGUIDSize attribute	6.2.21	M	
SCR-DS-DEVINF-C-022	Support for MaxID attribute	6.2.22	O	
SCR-DS-DEVINF-C-023	Support for MaxMem attribute	6.2.23	O	
SCR-DS-DEVINF-C-024	Support for MaxOccur element	6.2.24	O	
SCR-DS-DEVINF-C-025	Support for MaxSize element	6.2.25	O	
SCR-DS-DEVINF-C-026	Support for MaxStoredAnchors attribute	6.2.26	O	

SCR-DS-DEVINF-C-027	Support for Model element	6.2.27	M	
SCR-DS-DEVINF-C-028	Support for Truncate attribute	6.2.50	M	
SCR-DS-DEVINF-C-029	Support for OEM element	6.2.28	O	
SCR-DS-DEVINF-C-030	Support for ParamName attribute	6.2.29	O	
SCR-DS-DEVINF-C-031	Support for Property element	6.2.30	M	
SCR-DS-DEVINF-C-032	Support for PropInfo element	6.2.31	M	
SCR-DS-DEVINF-C-033	Support for PropName element	6.2.32	M	
SCR-DS-DEVINF-C-034	Support for PropParam element	6.2.33	O	
SCR-DS-DEVINF-C-035	Support for Rx element	6.2.34	M	
SCR-DS-DEVINF-C-036	Support for Rx-Pref element	6.2.35	M	
SCR-DS-DEVINF-C-037	Support for RxTx-CT element	0	M	
SCR-DS-DEVINF-C-038	Support for SharedMem attribute	0	O	
SCR-DS-DEVINF-C-039	Support for SourceRef element	6.2.38	M	
SCR-DS-DEVINF-C-040	Support for StoredAnchors element	6.2.39	O	
SCR-DS-DEVINF-C-041	Support for SupportAtomic attribute	6.2.40	O	
SCR-DS-DEVINF-C-042	Support for SupportEncryption attribute	6.2.41	O	
SCR-DS-DEVINF-C-043	Support for SupportFieldLevel attribute	6.2.42	O	
SCR-DS-DEVINF-C-044	Support for SupportHierarchicalSync attribute	6.2.43	O	
SCR-DS-DEVINF-C-045	Include SupportLargeObjs attribute	6.2.44	O	
SCR-DS-DEVINF-C-046	Support for SupportNumberOfChanges attribute	6.2.45	O	
SCR-DS-DEVINF-C-047	Support for SupportSequence attribute	6.2.46	O	
SCR-DS-DEVINF-C-048	Support for SupportSftDel attribute	6.2.47	O	
SCR-DS-DEVINF-C-049	Support for SwV element	6.2.48	M	
SCR-DS-DEVINF-C-050	Support for SyncCap element	6.2.49	M	
SCR-DS-DEVINF-C-051	Support for Tx element	6.2.51	M	
SCR-DS-DEVINF-C-052	Support for Tx-Pref element	6.2.52	M	
SCR-DS-DEVINF-C-053	Support for UTC attribute	6.2.53	O	
SCR-DS-DEVINF-C-054	Support for ValEnum element	6.2.54	M	
SCR-DS-DEVINF-C-055	Support for ValidAnchor element	6.2.55	O	
SCR-DS-DEVINF-C-056	Support for VerCT element	6.2.56	M	

SCR-DS-DEVINF-C-057	Support for Version attribute	6.2.57	M	
SCR-DS-DEVINF-C-058	Support for XName element	6.2.58	O	
SCR-DS-DEVINF-C-059	Support for XValue element	6.2.59	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	6.2.1	O	
SCR-DS-DEVINF-S-002	Support for CTType element	6.2.2	M	
SCR-DS-DEVINF-S-003	Support for DataStore element	6.2.3	M	
SCR-DS-DEVINF-S-004	Support for DataType element	6.2.4	M	
SCR-DS-DEVINF-S-005	Support for DevCap element	6.2.5	M	
SCR-DS-DEVINF-S-006	Support for DevID element	6.2.6	M	
SCR-DS-DEVINF-S-007	Support for DevInf element	6.2.7	M	
SCR-DS-DEVINF-S-008	Support for DevType element	6.2.8	M	
SCR-DS-DEVINF-S-009	Support for DisplayName attribute	6.2.9	O	
SCR-DS-DEVINF-S-010	Support for DSMem element	6.2.10	O	
SCR-DS-DEVINF-S-011	Support for Ext element	6.2.11	O	
SCR-DS-DEVINF-S-012	Support for ExtURI element	6.2.12	O	
SCR-DS-DEVINF-S-013	Support for FieldLevel attribute	6.2.13	O	
SCR-DS-DEVINF-S-014	Support for Filter-Rx element	6.2.16	O	
SCR-DS-DEVINF-S-015	Support for FilterCap element	6.2.14	O	
SCR-DS-DEVINF-S-016	Support for FPUnique attribute	6.2.17	O	
SCR-DS-DEVINF-S-017	Support for FilterKeyword element	6.2.15	O	
SCR-DS-DEVINF-S-018	Support for FwV element	6.2.18	M	
SCR-DS-DEVINF-S-019	Support for HwV element	6.2.19	M	
SCR-DS-DEVINF-S-020	Support for Man element	6.2.20	M	
SCR-DS-DEVINF-S-021	Support for MaxGUIDSize attribute	6.2.21	M	
SCR-DS-DEVINF-S-022	Support for MaxID attribute	6.2.22	O	
SCR-DS-DEVINF-S-023	Support for MaxMem attribute	6.2.23	O	
SCR-DS-DEVINF-S-024	Support for MaxOccur element	6.2.24	M	

SCR-DS-DEVINF-S-025	Support for MaxSize element	6.2.25	M	
SCR-DS-DEVINF-S-026	Support for MaxStoredAnchors attribute	6.2.26	O	
SCR-DS-DEVINF-S-027	Support for Model element	6.2.27	M	
SCR-DS-DEVINF-S-028	Support for Truncate attribute	6.2.50	M	
SCR-DS-DEVINF-S-029	Support for OEM element	6.2.28	O	
SCR-DS-DEVINF-S-030	Support for ParamName attribute	6.2.29	O	
SCR-DS-DEVINF-S-031	Support for Property element	6.2.30	M	
SCR-DS-DEVINF-S-032	Support for PropInfo element	6.2.31	M	
SCR-DS-DEVINF-S-033	Support for PropName element	6.2.32	M	
SCR-DS-DEVINF-S-034	Support for PropParam element	6.2.33	O	
SCR-DS-DEVINF-S-035	Support for Rx element	6.2.34	M	
SCR-DS-DEVINF-S-036	Support for Rx-Pref element	6.2.35	M	
SCR-DS-DEVINF-S-037	Support for RxTx-CT element	0	M	
SCR-DS-DEVINF-S-038	Support for SharedMem attribute	0	O	
SCR-DS-DEVINF-S-039	Support for SourceRef element	6.2.38	M	
SCR-DS-DEVINF-S-040	Support for StoredAnchors element	6.2.39	O	
SCR-DS-DEVINF-S-041	Support for SupportAtomic attribute	6.2.40	O	
SCR-DS-DEVINF-S-042	Support for SupportEncryption attribute	6.2.41	O	
SCR-DS-DEVINF-S-043	Support for SupportFieldLevel attribute	6.2.42	O	
SCR-DS-DEVINF-S-044	Support for SupportHierarchicalSync attribute	6.2.43	O	
SCR-DS-DEVINF-S-045	Include SupportLargeObjs attribute	6.2.44	M	
SCR-DS-DEVINF-S-046	Support for SupportNumberOfChanges attribute	6.2.45	M	
SCR-DS-DEVINF-S-047	Support for SupportSequence attribute	6.2.46	O	
SCR-DS-DEVINF-S-048	Support for SupportSftDel attribute	6.2.47	O	
SCR-DS-DEVINF-S-049	Support for SwV element	6.2.48	M	
SCR-DS-DEVINF-S-050	Support for SyncCap element	6.2.49	M	
SCR-DS-DEVINF-S-051	Support for Tx element	6.2.51	M	
SCR-DS-DEVINF-S-052	Support for Tx-Pref element	6.2.52	M	
SCR-DS-DEVINF-S-053	Support for UTC attribute	6.2.53	M	
SCR-DS-DEVINF-S-054	Support for ValEnum element	6.2.54	M	

SCR-DS-DEVINF-S-055	Support for ValidAnchor element	6.2.55	O	
SCR-DS-DEVINF-S-056	Support for VerCT element	6.2.56	M	
SCR-DS-DEVINF-S-057	Support for Version attribute	6.2.57	M	
SCR-DS-DEVINF-S-058	Support for XName element	6.2.58	O	
SCR-DS-DEVINF-S-059	Support for XValue element	6.2.59	O	

Appendix B. Change History (Informative)

B.1 Approved Version 2.0 History

Reference	Date	Description
OMA-TS-DS_DevInf-V2_0-20110719-A	19 Jul 2011	Status changed to Approved by TP: OMA-TP-2011-0258-INP_DS_V2_0_ERP_for_final_Approval