Folder data object specification
Candidate Version 1.2 – 16 Mar 2006

Open Mobile Alliance
OMA-TS-DS_DataObjFolder-V1_2-20060316-C
Contents

1. SCOPE ................................................................................................................................. 5
2. REFERENCES .......................................................................................................................... 6
  2.1 NORMATIVE REFERENCES .............................................................................................. 6
  2.2 INFORMATIVE REFERENCES ......................................................................................... 6
3. TERMINOLOGY AND CONVENTIONS .................................................................................... 7
  3.1 CONVENTIONS .................................................................................................................. 7
  3.2 DEFINITIONS ...................................................................................................................... 7
  3.3 ABBREVIATIONS ................................................................................................................. 7
4. INTRODUCTION ..................................................................................................................... 8
5. XML USAGE .......................................................................................................................... 9
  5.1 INCLUDING FOLDER OBJECT IN OTHER XML DOCUMENTS ........................................... 9
  5.2 XML NAMESPACES .......................................................................................................... 9
  5.3 XML ATTRIBUTES ............................................................................................................. 9
  5.4 WBXML ............................................................................................................................... 9
6. MIME USAGE ....................................................................................................................... 9
7. DATA TYPES ........................................................................................................................ 10
  7.1 DATETIME ........................................................................................................................ 10
  7.2 BOOL ................................................................................................................................... 10
  7.3 TEXT .................................................................................................................................. 11
  7.4 INT ....................................................................................................................................... 11
8. MARK-UP LANGUAGE DESCRIPTION .................................................................................. 13
  8.1 FOLDER ............................................................................................................................. 13
  8.2 NAME .................................................................................................................................. 13
  8.3 CREATED ............................................................................................................................ 13
  8.4 MODIFIED .......................................................................................................................... 13
  8.5 ACCESSED ........................................................................................................................ 13
  8.6 ATTRIBUTES ...................................................................................................................... 14
    8.6.1 h ...................................................................................................................................... 14
    8.6.2 s ...................................................................................................................................... 14
    8.6.3 a ...................................................................................................................................... 14
    8.6.4 d ...................................................................................................................................... 14
    8.6.5 w ...................................................................................................................................... 15
    8.6.6 r ...................................................................................................................................... 15
    8.6.7 x ...................................................................................................................................... 15
  8.7 ROLE ................................................................................................................................... 15
  8.8 EXTENSION FIELDS .......................................................................................................... 16
    8.8.1 Unique naming ............................................................................................................. 16
    8.8.2 Ext ................................................................................................................................... 16
    8.8.3 XNam ............................................................................................................................ 17
    8.8.4 XVal ............................................................................................................................... 17
9. DTD .......................................................................................................................................... 19
10. SYNCML DATA SYNCHRONIZATION USAGE ................................................................. 21
  10.1 CTCAP ............................................................................................................................. 21
  10.2 DATA SYNC RECORD AND FIELD LEVEL FILTERING ................................................... 21
    10.2.1 Folder Media Object Filter ......................................................................................... 21
  10.3 FOLDER OBJECT REPLACE EXAMPLE ........................................................................ 21
APPENDIX A CHANGE HISTORY (INFORMATIVE) ............................................................... 23
  A.1 APPROVED VERSION HISTORY ...................................................................................... 23
  A.2 DRAFT/CANDIDATE VERSION 1.2 HISTORY ................................................................... 23
## Tables

Table 1 Standard folder role identifiers

<table>
<thead>
<tr>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1 Standard folder role identifiers</td>
<td>16</td>
</tr>
</tbody>
</table>
1. Scope

The folder data object is presented in this document. The content-specific aspects of synchronization (filtering keywords, etc…) are listed and clarified.
2. References

2.1 Normative References


2.2 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.

3.2 Definitions

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

3.3 Abbreviations

OMA Open Mobile Alliance
4. Introduction

The folder data objects can be used to organize objects such as email, file, other folder objects, etc. in a hierarchical manner.
5. XML Usage

The *folder* objects are represented in a mark-up language defined by [XML]. The *folder* is an XML application. The *folder* DTD (Document Type Definition) defines the XML document type used to represent a *folder* object. The *folder* DTD can be found in Section 8.7, but it is not necessary to read the DTD in order to understand it.

*Folder* objects are specified using well-formed XML. However, the *folder* need not be valid XML. That is, the *folder* objects do not need to specify the XML declaration or prolog. They only need to specify the body of the XML document. This restriction allows for the *folder* objects to be specified with greater terseness than well-formed, valid XML documents.

5.1 Including folder object in other XML documents

When using the *folder* object as content of another XML document element the mark-up characters MUST be properly escaped or the CDATA sections MUST be used. See [XML] specification for details on character escaping and usage of CDATA sections. See section 10.3 of this document for an example.

5.2 XML Namespaces

*Folder* objects to date have no elements that may include elements of the other namespaces.

5.3 XML Attributes

In order to simplify the implementation of the *folder* objects in small devices, the *folder* objects have been intentionally designed to use the XML elements only. Currently no XML attributes are being defined for *folder* objects.

5.4 WBXML

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, *folder* uses shortened element type. This provides a minor reduction in verbosity.

Additionally, the *folder* objects can be encoded in a tokenized, binary format defined by [WBXML]. The use of [WBXML] format is external to specification of the *folder* and should be transparent to any application. The combination of the use of shortened element type names and an alternative binary format makes *folder* object competitive, from a compressed format perspective, with alternative, but private, binary representations.

6. MIME Usage

The [RFC2045] Internet standard provides an industry-accepted mechanism for identifying different content types. The *folder* object is identified by a MIME media type. The `application/vnd.omads-folder+xml` MIME content type MUST be used to indicate the *folder* object wherever such indication is required.
7. Data types

The following basic data type definitions are provided for referencing from other parts of this document.

7.1 datetime

Usage: This value type is used to identify values that specify a precise calendar date and time of day.

Description:

The datetime data type is used to identify values that contain a precise calendar date and time of day. The format is based on the [ISO 8601] complete representation, basic format for a calendar date and time of day. The text format is a concatenation of the "date", followed by the LATIN CAPITAL LETTER T character (US-ASCII decimal 84) time designator, followed by the "time" format.

The datetime data type expresses time values in two forms:

The form of date and time with UTC offset MUST NOT be used. For example, the following is not valid for a date-time value:

```
...<datefield>19980119T230000-0800</datefield> <!-- Invalid time format -->
...```

FORM #1: DATE WITH LOCAL TIME

The date with local time form is simply a date-time value that does not contain the UTC designator nor does it reference a time zone. For example, the following represents January 18, 1998, at 11 PM:

```
...<datefield>19980118T230000</datefield> <!-- January 18, 1998, 11 PM -->
...```

This notation of datetime type is to be used by devices that have no knowledge of the time zone in which they operate. In this case, the datetime value that is being transferred is usually the same as the value that is being stored and shown to the user in the application UI.

FORM #2: DATE WITH UTC TIME

The date with UTC time, or absolute time, is identified by a LATIN CAPITAL LETTER Z suffix character (US-ASCII decimal 90), the UTC designator, appended to the time value. For example, the following represents January 19, 1998, at 0700 UTC:

```
...<datefield>19980119T070000Z</datefield> <!-- January 19,1998,07:00 UTC -->
...```

7.2 bool

Usage: To be used for Boolean type fields
Restrictions: A text value that MUST be either “true” to indicate Boolean “true” or “false” to indicate “false”. If the field is not present its value is assumed to be “false”.

Example:

```xml
...<booleanfield>true</booleanfield> <!-- the field is set to “true” --> ...
```

7.3 text

Usage: To be used for textual fields

Restrictions: If the field is not present its value is assumed to be an empty string.

Example:

```xml
...<textfield>Hello World!</textfield> ...
```

7.4 int

Usage: To be used for integer numeric fields.

Restrictions: The format for the integer values is defined here in an ABNF notation [RFC2234].

```plaintext
nonzero-digit = "1"/ "2"/ "3"/ "4"/ "5"/ "6"/ "7"/ "8"/ "9"

octal-digit = "0"/ "1"/ "2"/ "3"/ "4"/ "5"/ "6"/ "7"

hexadecimal-digit = "0"/ "1"/ "2"/ "3"/ "4"/ "5"/ "6"/ "7"/ "8"/ "9" / "a"/ "b"/ "c"/ "d"/ "e"/ "f" / "A"/ "B"/ "C"/ "D"/ "E"/ "F"

decimal-constant = nonzero-digit *("0" / nonzero-digit)

hexadecimal-constant = ("0x" / "0X") 1*hexadecimal-digit
```
octal-constant = "0" 1*octal-digit

integer-value = *1("+" / "-") (decimal-constant / hexadecimal-constant / octal-constant)

Example:

...  
<negativevalue>-1234</negativevalue>  
<positivevalue>1234</positivevalue>  
<anotherpositivevalue>+0xfffabc5</anotherpositivevalue>  
<octal>010</octal> <!-- octal value equivalent to decimal 8 -->  
...
8. Mark-up Language Description

8.1 Folder

Usage: Indicates the beginning of the object

Parent elements: None

Content model:

Folder (name?, created?, modified?, accessed?, attributes?, role?, Ext*)

8.2 name

Usage: Specifies name of the folder.

Parent elements: Folder

Restrictions: The supporters of the folder object MUST support this property. The name does not include any location information (URL, path, etc.). The name of the folder must not be empty.

Content model:

name (#PCDATA)

8.3 created

Usage: Specifies the date and time when the folder was created

Parent elements: Folder

Restrictions: datatime type field as specified in section 7.1.

Content model:

created (#PCDATA)

8.4 modified

Usage: Specifies the date and time when the folder object was last changed.

Parent elements: Folder

Restrictions: datatime type field as specified in section 7.1.

Content model:

modified (#PCDATA)

8.5 accessed

Usage: Specifies the date and time when folder object was last accessed.

Parent elements: Folder
Restrictions: *datetime* type field as specified in section 7.1.

Content model:

```
accessed (#PCDATA)
```

## 8.6 attributes

Usage: Specifies the state of the file system attributes of the file.

**Parent elements:** Folder

**Restrictions:**

Content model:

```
attributes (h?, s?, a?, d?, w?, r?, x?)
```

### 8.6.1 h

Usage: Specifies “hidden” attribute state.

**Parent elements:** attributes

**Restrictions:** *bool* type field as specified in section 7.2.

Content model:

```
h (#PCDATA)
```

### 8.6.2 s

Usage: Specifies “system” attribute state.

**Parent elements:** attributes

**Restrictions:** *bool* type field as specified in section 7.2.

Content model:

```
s (#PCDATA)
```

### 8.6.3 a

Usage: Specifies “archived” attribute state.

**Parent elements:** attributes

**Restrictions:** *bool* type field as specified in section 7.2.

Content model:

```
a (#PCDATA)
```

### 8.6.4 d

Usage: Specifies “delete” attribute state. When the “delete” attribute is set, the file can be deleted.
8.6.5  \textit{w}

Usage: Specifies “writable” attribute state. If the “writable” attribute is set, the file can be written.

Parent elements: attributes

Restrictions: bool type field as specified in section 7.2.

Content model:

\begin{verbatim}
 w (#PCDATA)
\end{verbatim}

8.6.6  \textit{r}

Usage: Specifies “readable” attribute state. If the “readable” attribute is set, it is possible to read the contents of the file.

Parent elements: attributes

Restrictions: bool type field as specified in section 7.2.

Content model:

\begin{verbatim}
 r (#PCDATA)
\end{verbatim}

8.6.7  \textit{x}

Usage: Specifies “executable” attribute state. If the “executable” attribute is set, the contents of the file can be executed by the operating system.

Parent elements: attributes

Restrictions: bool type field as specified in section 7.2.

Content model:

\begin{verbatim}
 x (#PCDATA)
\end{verbatim}

8.7  \textbf{role}

Usage: Identifies the role of the folder within the store. Applications are expected to use this field when it is required to establish mapping between folders of different data stores (e.g. when two stores are being synchronized).

Parent elements: Folder

Restrictions: The following standard folder role identifiers can be used:
Table 1 Standard folder role identifiers

<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Message folder roles</strong></td>
<td></td>
</tr>
<tr>
<td>Inbox</td>
<td>The default folder for the delivered inbound messages</td>
</tr>
<tr>
<td>Outbox</td>
<td>The default folder for the messages that are pending the delivery</td>
</tr>
<tr>
<td>Drafts</td>
<td>The default folder for the messages that are being edited by the user.</td>
</tr>
<tr>
<td>Sent</td>
<td>The default folder for delivered outbound messages.</td>
</tr>
<tr>
<td><strong>File folder roles</strong></td>
<td></td>
</tr>
<tr>
<td>Documents</td>
<td>The default location for documents</td>
</tr>
<tr>
<td>Pictures</td>
<td>The default location for still pictures</td>
</tr>
<tr>
<td>Movies</td>
<td>The default location for movies</td>
</tr>
<tr>
<td>Music</td>
<td>The default location for music files</td>
</tr>
<tr>
<td>Applications</td>
<td>The default location of application files</td>
</tr>
</tbody>
</table>

In addition to standard identifiers, vendor-specific identifiers can be used. Refer to section 8.8.1 for guidelines on identifier naming.

## 8.8 Extension fields

### 8.8.1 Unique naming

If an extension field is required, the following naming convention MUST be followed in order to prevent undesirable field name collisions.

\[
x\text{-name} = "x-" \text{vendorid } "-" 1*(ALPHA / DIGIT / "-") \quad ; \text{field name}
\]

\[
\text{vendorid} = 3*(ALPHA / DIGIT) \quad ; \text{Vendor identification}
\]

\[
\text{ALPHA} = \%\text{x41-5A} / \%\text{x61-7A} \quad ; \text{A-Z / a-z}
\]

\[
\text{DIGIT} = \%\text{x30-39} \quad ; \text{0-9}
\]

### 8.8.2 Ext

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

**Parent Elements:** Folder
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:

```xml
Ext (XNam, XVal*)
```

Attributes: None.

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

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

8.8.3 XNam

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

Parent Elements: Ext

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

Content Model:

```xml
XNam (#PCDATA)
```

Attributes: None.

Example:

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

8.8.4 XVal

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

Parent Elements: Ext

Restrictions:

Content Model:

```xml
XVal (#PCDATA)
```
Attributes: None.

Example:

```xml
<Ext>
  <XNam>x-Foo-CliVer</XNam>
  <XVal>5.0</XVal>
  <XVal>5.01</XVal>
  <XVal>5.02</XVal>
</Ext>
```
9. DTD

```xml
<!--
application/vnd.omads-folder+xml V1.2 Document Type Definition

http://www.openmobilealliance.org/tech/DTD/OMA-DS-DataObjFolder-DTD-V1_2.dtd

Copyright Open Mobile Alliance Ltd., 2002-2003
All rights reserved
Terms and conditions of use are available from the
Open Mobile Alliance Ltd. web site at
http://www.openmobilealliance.org/useterms.html -->

<?xml version="1.0" encoding="UTF-8"?>
<!-- Root Element -->
<!ELEMENT Folder (name?, created?, modified?, accessed?, attributes?,
role?, Ext*)>

<!--: The supporters of the folder object MUST support this property. The
name does not include any location information (URL, path, etc.). -->
<!ELEMENT name (#PCDATA)>
<!ELEMENT created (#PCDATA)>
<!ELEMENT modified (#PCDATA)>
<!ELEMENT accessed (#PCDATA)>
<!ELEMENT attributes (h?, s?, a?, d?, w?, r?, x?)>
<!ELEMENT h (#PCDATA)>
<!ELEMENT s (#PCDATA)>
<!ELEMENT a (#PCDATA)>
<!ELEMENT d (#PCDATA)>
<!ELEMENT w (#PCDATA)>
<!ELEMENT r (#PCDATA)>
<!ELEMENT x (#PCDATA)>
```
<!ELEMENT role (#PCDATA)>

<!ELEMENT Ext (XNam, XVal*)>

<!ELEMENT XNam (#PCDATA)>

<!ELEMENT XVal (#PCDATA)>

<!-- End of DTD Definition -->
10. SyncML Data Synchronization Usage

The following sections describe the content-specific recommendations for using the data synchronization [DSREPU] protocol with folder data objects.

10.1 CTCap

Refer to [DSDEVDTD] for further details on the specification of the Device Information DTD.

```xml
<CTCap>
  <CTType>application/vnd.omads-folder+xml</CTType>
  <Property>
    <PropName>name</PropName>
    <DataType>text</DataType>
    <MaxSize>260</MaxSize>
    <DisplayName>Folder name</DisplayName>
  </Property>
  <Property>
    <PropName>created</PropName>
    <DataType>datetime</DataType>
    <DisplayName>Date created</DisplayName>
  </Property>
</CTCap>
```

10.2 Data Sync Record and Field Level Filtering

10.2.1 Folder Media Object Filter

Filtering for file objects can be specified using both Record and Field elements. The set of recommended keywords to support is as follows:

```plaintext
cf-filter-keyword = <Any field that is defined for the application/vnd.omads-folder+xml content type in this document>
```

10.3 Folder object replace example

```xml
...<Sync>
```
...<Replace>
   <CmdID>6</CmdID>
   <Meta>
      <Type xmlns='syncml:metinf'>application/vnd.omadsfolder+xml</Type>
   </Meta>
   <Item>
      <Source>
         <LocURI>123</LocURI>
      </Source>
      <SourceParent>
         <LocURI>345</LocURI>
      </SourceParent>
      ...
      <Data><![CDATA[
         <Folder>
            <name>OMAFiles</name>
            <created>20030807T231830</created>
         </Folder>]]>
      </Data>
      </Item>
   </Replace>
</Sync>
## Appendix A  Change History

### A.1  Approved Version History

<table>
<thead>
<tr>
<th>Reference</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>n/a</td>
<td>n/a</td>
<td>No prior version – or - No previous version within OMA</td>
</tr>
</tbody>
</table>

### A.2  Draft/Candidate Version 1.2 History

<table>
<thead>
<tr>
<th>Document Identifier</th>
<th>Date</th>
<th>Sections</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OMA-SyncML-DataObjFolder-V1_2</td>
<td>19 Jan 04</td>
<td>Title page, p.2</td>
<td>Changed dates and copyrights</td>
</tr>
<tr>
<td></td>
<td>26 Jan 04</td>
<td>Last page, All</td>
<td>Removed Appendix B, corrected typos</td>
</tr>
<tr>
<td></td>
<td>01 Mar 04</td>
<td>TP, 2.1</td>
<td>Clerical changes from consistency review</td>
</tr>
<tr>
<td></td>
<td>14 Apr 04</td>
<td>5.1, 6, 9, 10.1, 10.2</td>
<td>Updated XML usage. Updated MIME types.</td>
</tr>
<tr>
<td></td>
<td>05 May 04</td>
<td>TP</td>
<td>Updated date</td>
</tr>
<tr>
<td>Candidate Version OMA-SyncML-DataObjFolder-V1_2</td>
<td>01 Jun 04</td>
<td>n/a</td>
<td>Status changed to Candidate by TP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TP ref # OMA-TP-2004-0178-DS-V1_2-for-candidate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Filename and name of specification updated All</td>
<td>Template update</td>
</tr>
</tbody>
</table>