TECHNICAL REPORT

ISO/IEC TR 30114-1

First edition
2016-12-15

Information technology — Extensions of Office Open XML file formats —

Part 1: Guidelines

Technologies de l'information — Extensions de formats de fichiers Office Open XML —
Partie 1: Lignes directrices

Reference number
ISO/IEC TR 30114-1:2016(E)

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see the following URL: www.iso.org/iso/foreword.html.

The committee responsible for this document is ISO/IEC JTC 1, *Information technology*, SC 34, *Document description and processing languages*.

A list of all parts in the ISO/IEC 30114 series, published under the general title *Information technology — Extensions of Office Open XML file formats*, can be found on the ISO website.

Introduction

ISO/IEC 29500 was designed to allow the addition of markup and other data to Office Open XML (OOXML) documents, and to allow OOXML applications unaware of such markup and date to provide reasonable results. ISO/IEC TR 30114-1 provides guidance for such additions, and also specifies a collection of such additions.

Information technology — Extensions of Office Open XML file formats —

Part 1: Guidelines

1 Scope

This document gives guidelines for the use of extensibility mechanisms in ISO/IEC 29500 (Office Open XML). In particular, it makes clear which of these mechanisms supports lossless round tripping.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

There are no normative references in this document.

3 Terms and definitions

No terms and definitions are listed in this document.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— IEC Electropedia: available at http://www.electropedia.org/

— ISO Online browsing platform: available at http://www.iso.org/obp

4 Adding markup or other data to OOXML documents

4.1 General

There are two main ways to add extra markup or other data to Office Open XML (OOXML) documents:

— Using the extension mechanisms described in ISO/IEC 29500-3, Markup Compatibility and Extensibility (MCE) offers three primary mechanisms for extending XML files, each with its own advantages and disadvantages.

— Embedding foreign Open Packaging Conventions (OPC) parts.

4.2 Markup Compatibility and Extensibility (MCE): Ignorable elements and attributes (ISO/IEC 29500-3)

The most commonly used extension mechanism, marking elements or attributes as ignorable, allows lightweight additions to be made to existing markup.

A good use of ignorable markup would be the addition of a custom metadata tag onto a paragraph in a WordprocessingML document. This could be accomplished by declaring a custom namespace, marking it as ignorable, and adding the attribute to the p element in that namespace. The relevant portions of the resulting document.xml part might resemble the following:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<w:document xmlns:mc=
  "http://schemas.openxmlformats.org/markup-compatibility/2006"
  xmlns:w="http://schemas.openxmlformats.org/wordprocessingml/2006/main"
  xmlns:mymeta="http://mywordprocessorapp.com/metadata" mc:Ignorable="mymeta">
  <w:body>
    <w:p mymeta:tag="marketing_team" w:rsidR="00AD3E96"
      w:rsidRDefault="00120C37">
      <w:r>
        <w:t>hello</w:t>
      </w:r>
    </w:p>
  </w:body>
</w:document>
```

Ignorable markup can be used anywhere in XML parts and requires minimal markup. It allows custom markup to be added to documents while retaining the document's conformance with the standard and allowing it to be opened by a third-party application without errors. However, ignorable elements and attributes will almost definitely be lost if files are round tripped (i.e., opened and saved again) in an application that does not understand them, as there is no requirement for applications to persist ignorable markup, and typically unknown ignorable markup is stripped during file load.

4.3 MCE: Alternate Content Blocks (ISO/IEC 29500-3)

While ignorable constructs allow markup to be added to documents easily, Alternate Content Blocks (ACBs) allow existing markup to be replaced, with the replacement targeted at particular consumers that understand it.

A good use of ACBs would be in developing an application that preferred to use the Open Document Format (ODF) in WordprocessingML paragraph markup. When creating files, the application would continue to write OpenXML markup in order to be compliant to the standard, but would also provide ODF markup in an ACB. When opening files, the application would disregard the OOXML fallback markup and only read the ODF.

The resulting document.xml part for a document with text stored in such a way might appear as shown below (note that, for simplicity, all ODF namespaces are merged into one):

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<w:document xmlns:mc=
  "http://schemas.openxmlformats.org/markup-compatibility/2006"
  xmlns:w="http://schemas.openxmlformats.org/wordprocessingml/2006/main"
  xmlns:myodf="http://mywordprocessorapp.com/odfcontent" mc:Ignorable="myodf">
  <w:body>
    <w:p w:rsidR="00AD3E96" w:rsidRDefault="00120C37">
      <mc:AlternateContent>
        <mc:Choice Requires="myodf">
          <myodf:style myodf:name="T2"
            myodf:parent-style-name="DefaultParagraphFont" myodf:family="text">
            <myodf:text-properties myodf:font-style="italic"
              myodf:font-style-asian="italic"/>
          </myodf:style>
          <myodf:p>This document is stored in<myodf:s/>
          <myodf:span myodf:style-name="T2">two</myodf:span>
          <myodf:s/>formats.</myodf:p>
        </mc:Choice>
        <mc:Fallback>
          <w:r>
            <w:t xml:space="preserve">This document is stored in </w:t>
          </w:r>
          <w:r>
            <w:rPr>
              <w:i/>
            </w:rPr>
            <w:t>two</w:t>
          </w:r>
          <w:r>
```



```
            <w:t xml:space="preserve"> formats.</w:t>
          </w:r>
        </mc:Fallback>
      </mc:AlternateContent>
    </w:p>
  </w:body>
</w:document>
```

ACBs allow for the replacement of existing markup for consumers that understand it. Much like ignorables, there is no requirement for applications to preserve ACBs on round-tripping operations, so data can be lost if third-party applications are used to open and save files.

4.4 MCE: Application-defined extension elements (ISO/IEC 29500-3)

Application-defined extension elements essentially allow markup designers to put "this space left for future expansion" elements into their formats. Syntactically, these are similar to ignorable elements but, because they only appear at predefined locations, markup consumers can easily keep track of unknown extension elements, which makes round tripping a simpler proposition.

In ISO/IEC 29500-1, SpreadsheetML is the only markup that utilizes extension elements (see extLst in ISO/IEC 29500-1). extLst elements occur at several predefined points in SpreadsheetML and allow the markup to be extended in a manner that permits round tripping.

A good use of SpreadsheetML's extension elements would be a spreadsheet application whose developers wished to add the ability for cells to be denoted as model inputs or outputs. Such an application could use these tags at runtime and, if users were to round trip the files in other applications, the markup would be preserved.

The CT_Cell type in SpreadsheetML contains an extLst element, so this will be an acceptable extension point. It contains an unbounded collection of ext elements, and the developer can add an ext with the developer's extension's markup. The resulting sheetData for a given spreadsheet might look something like the following:


```
<sheetData>
  <row r="1" spans="1:1" x14ac:dyDescent="0.25">
    <c r="A1">
      <v>1234</v>
      <extLst>
        <ext xmlns:mymodel="http://myspreadsheetapp.com/modelInputsAndOutputs"
          uri="http://myspreadsheetapp.com/modelInputsAndOutputs">
          <mymodel:modelSpecifier cellType="input" name="interestRate">
            <mymodel:description>Specifies the interest rate to be passed into
amortisation model.</mymodel:description>
          </mymodel:modelSpecifier>
        </ext>
      </extLst>
    </c>
  </row>
</sheetData>
```

Because consuming spreadsheet applications understand that this data is attached at a cell level, this metadata remains with the cell when it is moved around the sheet via cut/paste or through row/column insertion or deletions above it. Note that some implementations can parse through application-defined-extension elements and modify constructs within them – Microsoft Excel, for example, will look for any sqref elements in the namespace http://schemas.microsoft.com/office/excel/2006/main. It assumes that they will contain spreadsheet row/column references and adjusts them appropriately if that referenced cell area is moved around at runtime.

Application-defined extension elements are only usable in locations pre-defined by a markup language, but allow for data preservation in round-tripping scenarios.