INTERNATIONAL STANDARD

ISO/IEC
26300

First edition
2006-12-01
AMENDMENT 1
2012-03-15

Information technology — Open Document Format for Office Applications (OpenDocument) v1.0

AMENDMENT 1: Open Document Format for Office Applications (OpenDocument) v1.1

Technologies de l'information — Format de document ouvert pour applications de bureau (OpenDocument) v1.0

AMENDEMENT 1: Format de document ouvert pour applications de bureau (OpenDocument) v1.1

© ISO/IEC 2012

iTeh STANDARD PREVIEW
(standards.iteh.ai)

ISO/IEC 26300:2006/Amd 1:2012
https://standards.iteh.ai/catalog/standards/sist/f2ebca8c-c9a3-4b67-9802-
f9a188c91c04/iso-iec-26300-2006-amd-1-2012

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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

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.

Amendment 1 to ISO/IEC 26300:2006 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 34, *Document description and processing languages*.

This Amendment should be read in conjunction with ISO/IEC 26300:2006 and the associated Technical Corrigenda 1 and 2. The current edition of ISO/IEC 26300 should be understood by first applying the change specified in the Technical Corrigenda, then applying the changes specified in this Amendment.

Notation conventions

The title of each change is the complete reference to the section or sub-section being modified. In all cases (except for an amendment inserting two new Appendices) the title begins with the section or sub-section number, the section or sub-section name, and the page number. In most cases the location of the amendment is more precisely indicated by a paragraph number or other indicator. Paragraph numbers are determined by counting the number of paragraphs from the beginning of the section or sub-section in question, ignoring lists, tables and examples.

In the case of amendments to the normative Relax-NG schema, the lines of the schema that are amended are indicated in italics below the heading. Schema line numbers refer to the *amended* schema. Other guidance on the intended application of an amendment may be given in italics below the heading.

A change can contain any one or more of the following kinds of edits:

Addition of text: New text is displayed in blue and is underlined, as demonstrated here.

Deletion of text: Deleted text is displayed in red and is struck-through, as demonstrated here.

An ellipsis '…' is occasionally used to indicate deliberate omission of fragments of the original text that are unchanged by this Amendment and would unreasonably extend the text of this Amendment.

Front matter, p. 1

Open Document Format for Office Applications (OpenDocument) v1.1~~0~~

~~(Second Edition)~~

~~Committee Specification1, 19 Jul 2006~~OASIS Standard, 1 Feb 2007

~~**Document identifier:**~~
~~OpenDocument-v1.1ed2-.odt~~

~~**Location:**~~
~~This Version: http://www.oasis-open.org/committees/office~~
~~Previous Version: http://docs.oasis-open.org/office/v1.0~~

Specification URIs

This Version:
http://docs.oasis-open.org/office/v1.1/OS/OpenDocument-v1.1.odt
http://docs.oasis-open.org/office/v1.1/OS/OpenDocument-v1.1.pdf
http://docs.oasis-open.org/office/v1.1/OS/OpenDocument-v1.1.html.zip

Previous Version:
http://www.oasis-open.org/committees/download.php/19275/OpenDocument-v1.0ed2-cs1.odt
http://www.oasis-open.org/committees/download.php/19274/OpenDocument-v1.0ed2-cs1.pdf

Latest Version:
http://docs.oasis-open.org/office/v1.1/OpenDocument-v1.1.odt
http://docs.oasis-open.org/office/v1.1/OpenDocument-v1.1.pdf
http://docs.oasis-open.org/office/v1.1/OpenDocument-v1.1.html.zip

Latest Approved Version:
http://docs.oasis-open.org/office/v1.1/OS/OpenDocument-v1.1.odt
http://docs.oasis-open.org/office/v1.1/OS/OpenDocument-v1.1.pdf
http://docs.oasis-open.org/office/v1.1/OS/OpenDocument-v1.1.html.zip

Technical Committee:
OASIS Open Document Format for Office Applications (OpenDocument) TC

Chair:
Michael Brauer, Sun Microsystems, Inc.

Editors:
Patrick Durusau, Individual
Michael Brauer, Sun Microsystems, Inc.
Lars Oppermann, Sun Microsystems, Inc.

Related Work:
This specification supersedes OASIS OpenDocument v1.0.

Declared XML Namespaces:
A list of XML namespaces declared by this specification is available in section 1.3.

Abstract:
This is the specification of the Open Document Format for Office Applications (OpenDocument) format, an open, XML-based file format for office applications, based on OpenOffice.org XML [OOo].

Status:

This document was last revised or approved by the ~~membership of OASIS~~OASIS Open Document Format for Office Applications (OpenDocument) Technical Committee on the above date. The level of approval is also listed above. Check the current location noted above for possible later revisions of this document. This document is updated periodically on no particular schedule.

Technical Committee members should send comments on this specification to the Technical Committee's email list. Others should send comments to the Technical Committee by using the "Send A Comment" button on the Technical Committee's web page at

www.oasis-open.org/committees/office.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page

(www.oasis-open.org/committees/office/ipr.php.

The non-normative errata page for this specification is located at www.oasisopen.org/committees/office.

§ 1.4, "Relax-NG Schema", schema fragment, p. 32

{The following text amends lines 3-4 and 9-10 of the normative Relax-NG schema, as reproduced in this section.}

```
1   <?xml version="1.0" encoding="UTF-8"?>
2   <!--
3       OASIS OpenDocument v1.10 (second edition)
4       OASIS Standard, 1 Feb 2007Committee Specification1, 19 Jul 2006
5       Relax-NG Schema
6
7       $Id$
8
9       © 2002-20075 OASIS Open
10      © 1999-20075 Sun Microsystems, Inc.
11  -->
```

§ 1.5, "Document Processing and Conformance", paragraph 7, p. 34

There are no rules regarding the elements and attributes that actually have to be supported by conforming applications, except that applications should not use foreign elements and attributes for features defined in~~by~~ the OpenDocument schema. See also appendix D.

§ 2.1.1, "Document Root Element Content Models", paragraph 1, p. 37

{The following text corrects an editorial error in ISO/IEC 26300:2006/Cor 1, in which part of the text of the sentence inserted at the end of the paragraph is missing.}

The content models of the five root elements is summarized in the following table. Note that `<office:document>` may contain all supported top-level elements. All four subdocument root elements together contain the same information as an <u>`<office:document>`</u> element that contains the same subdocument root elements.

§ 2.1.1, table following paragraph 1, p. 37

{The following text amends the heading of the second column of the table.}

meta-
data

§ 2.3.1, "Text Documents", sub-section "Text Document Content Model", schema fragment, p. 42

{The following text inserts a new line (line 179 of the amended schema) in the definition of pattern "text-content" in the normative Relax-NG schema, as reproduced in this section.}

```
170  <define name="text-content">
171      <choice>
172          <ref name="text-h"/>
173          <ref name="text-p"/>
174          <ref name="text-list"/>
175          <ref name="text-numbered-paragraph"/>
176          <ref name="table-table"/>
177          <ref name="draw-a"/>
178          <ref name="text-section"/>
179          <ref name="text-soft-page-break"/>
180          <ref name="text-table-of-content"/>
181          <ref name="text-illustration-index"/>
182          <ref name="text-table-index"/>
183          <ref name="text-object-index"/>
184          <ref name="text-user-index"/>
185          <ref name="text-alphabetical-index"/>
186          <ref name="text-bibliography"/>
187          <ref name="shape"/>
188          <ref name="change-marks"/>
189      </choice>
190  </define>
```

§ 2.3.1, "Text Documents", following sub-section "Global Text Documents", p. 42

{The following text inserts a new sub-section, including reproduction of a new pattern definition "office-text-attlist" inserted into the normative Relax-NG schema (lines 201-207 of the amended schema).}

Use Soft Page Breaks

The `text:use-soft-page-breaks` attribute specifies whether the document contains soft page breaks.

A soft page break is a page break that has been included by a page oriented processor at a position where the document itself does not include a page break (e.g. by using the `fo:break-before` and `fo:break-after` formatting properties described in section 15.5.2).

Soft page breaks are specified by the `<text:soft-page-break>` elements described in sections 4.7 and 5.1.1:Soft Page breaks.

The use of the `<text:soft-page-break>` elements is always optional. An application generating the format **may** include the element if it has computed a paginated layout. A consuming application **may** handle the element while computing the layout, but it **shall not** depend on its existence. Soft page breaks are only supported within text documents.

A generating application that stores soft page breaks **shall** indicate this by setting the `text:use-page-breaks` attribute to `true`. A generating application that does not store soft page breaks **shall** indicate that by omitting this attribute, or by setting it to `false`.

An application that does not support pagination and soft page-breaks, that modifies an OpenDocument file, which includes soft page-breaks, **shall** at least set the `text:use-page-breaks` attribute to `false` (or remove it). It **should** also remove the `<text:soft-page-break>` elements from the document but is not required to do so.

An application that computes a paginated layout of a document **should** provide a facility to turn on export of soft page breaks for the purposes of consistent page breaks and for proper conversion to digital talking book formats (such as [DAISY]).

For `<text:soft-page-break>` elements that appear within table rows, the maximum number of `<text:soft-page-break>` elements that appear within the single table cells determines the number of page breaks that appear within the table row. The `<text:soft-page-break>` elements contained in each cell determine the positions where these page breaks appear within the cell content.

Similarly, `<text:soft-page-break>` elements that appear within text boxes and other content displayed outside the text flow, do not start a new page, but only indicate where the text-box's content breaks between two pages.

```
201   <define name="office-text-attlist" combine="interleave">
202       <optional>
203           <attribute name="text:use-soft-page-breaks" a:defaultValue="false">
204               <ref name="boolean"/>
205           </attribute>
206       </optional>
207   </define>
```

§ 2.5, "Scripts", paragraph 2, p. 50

Scripts do not imply a scripting language or an object model. A script can for instance operate on the Document Object Model (DOM) composed from the XML representation of a document in OpenDocument format (see [DOM2]), operate on the Document Object Model (DOM) of a document in OpenDocument format or on an application specific API.

§ 3, "Meta Data Elements", section title, p. 56

{The section title is revised as follows.}

3 Metad Data Elements

§ 3.1.18, "Document Statistics", schema fragment, pp. 62–63

{The following text moves five lines of the definition of the pattern "office-meta-data" to a new position (lines 771-775 of the amended schema) in the normative Relax-NG schema, as reproduced in this section.}

```
744   <define name="office-meta-data" combine="choice">
745       <element name="meta:document-statistic">
              ...
766           <optional>
767               <attribute name="meta:ole-object-count">
768                   <ref name="nonNegativeInteger"/>
769               </attribute>
770           </optional>
771           <optional>
772               <attribute name="meta:object-count">
773                   <ref name="nonNegativeInteger"/>
774               </attribute>
775           </optional>
```

```
776        <optional>
777            <attribute name="meta:paragraph-count">
778                <ref name="nonNegativeInteger"/>
779            </attribute>
780        </optional>
           ...
816        <optional>
817            <attribute name="meta:cell-count">
818                <ref name="nonNegativeInteger"/>
819            </attribute>
820        </optional>
           <optional>
               <attribute name="meta:object-count">
                   <ref name="nonNegativeInteger"/>
               </attribute>
           </optional>
821    </element>
822 </define>
```

§ 4.3.2, "List Item", paragraph 1, p. 71

List items contain the textual content of a list. A `<text:list-item>` element can contain paragraphs, headings, lists or soft page breaks. A list item cannot contain or lists. A list item cannot contain headings or tables.

§ 4.3.2, "List Item", pp. 71-72

{The following text inserts a new line in the definition of the pattern "text-list-item-content" (line 994 of the amended schema) in the normative Relax-NG schema, as reproduced in this section.}

```
979 <define name="text-list-item">
980    <element name="text:list-item">
981        <ref name="text-list-item-attlist"/>
982        <ref name="text-list-item-content"/>
983    </element>
984 </define>
985 <define name="text-list-item-content">
986    <optional>
987        <ref name="text-number"/>
988    </optional>
989    <zeroOrMore>
990        <choice>
991            <ref name="text-p"/>
992            <ref name="text-h"/>
993            <ref name="text-list"/>
994            <ref name="text-soft-page-break"/>
995        </choice>
996    </zeroOrMore>
997 </define>
```

§ 4.3.2, "List Item", sub-section "Formatted number", example, p. 72

```
<text:list text:style-name="List 1">
  <text:list-item>
  <text:p>This is the first list item</text:p>
  <text:p>This is a continuation of the first list item.</text:p>
  </text:list-item>
  <text:list-item>
  <text:p>This is the second list item.
         It contains a sub list.</text:p>
```

5

```
    <text:list>
        <text:list-item><text:p>This is a sub list item.</text:p>
        </text:list-item>
        <text:list-item><text:p>This is a sub list item.</text:p>
        </text:list-item>
        <text:list-item><text:p>This is a sub list item.</text:p>
        </text:list-item>
    </text:list>
    </text:list-item>
    <text:list-item>
    <text:p>This is the third list item</text:p>
    </text:list-item>
</text:list>
```

§ 4.4.3, "DDE Source", following paragraph 1, p. 77

{The following text inserts a new paragraph following existing paragraph 1 and preceding the schema fragment.}

See section 12.6 for the use of DDE connections.

§ 4.6.4, "Deletion", second bulleted list, p. 80

- If the change mark is inside a paragraph, insert the text content of the <text:deletion> element as if the beginning <text:p> and final </text:p> tags were missing.

- If the change mark is inside a header, proceed as above; except adapt the end tags to match~~inserted tags to math~~ their new counterparts.

- Otherwise, simply copy the text content of the <text:deletion> element in place of the change mark.

§ 4.6.4, "Deletion", example, pp. 80-81

Example: Given the following change:

...

This becomes:

```
<text:p>abcdef</text:p>
<text:p>Hello</text:p><text:h>Hello</text:h>
<text:p>World!</text:p><text:h>World!</text:h>
<text:p>ghijkl</text:p>
```

If, in the first two cases, the deletion contains complete paragraphs, then additional empty paragraphs must be put into the <text:deletion> element to achieve the desired result.

...

would be represented as:

```
<text:changed-region text:id="example">
    <text:deletion>
        <office:change-info>...</office:change-info>
        <text:p/>
        <text:h>Hello</text:h><text:p>Hello</text:p>
        <text:h>World!</text:h><text:p>World!</text:p>
        <text:p/>
    </text:deletion>
</text:changed-region>
```

New section following § 4.6 "Change Tracking", p. 82

{The following text inserts a new sub-section, including reproduction of a new pattern definition "text-soft-page-break" (lines 1199-1203 of the amended schema) inserted into the normative Relax-NG schema.}

4.7 Soft Page Break

The `<text:soft-page-break>` element represents a soft page break.

See section 2.3.1:Use Soft Page Breaks for details regarding soft page breaks.

```
1199   <define name="text-soft-page-break">
1200      <element name="text:soft-page-break">
1201         <empty/>
1202      </element>
1203   </define>
```

§ 5.1.1, "White-space Characters", paragraphs 1-3, p. 84

{The following text amends paragraphs 1 and 3 and inserts a new paragraph following existing paragraph 3.}

If the paragraph element or any of its child elements contains white-space characters, they are collapsed. Leading white-space characters at the paragraph start as well as trailing white-space characters at the paragraph end are ignored. In detail, the following conversions take place, in other words they are processed in the same way that [HTML4] processes them. The following [UNICODE] characters are normalized to a SPACE character:

The following [UNICODE] characters are normalized to a SPACE character:

- HORIZONTAL TABULATION (0x0009)

- CARRIAGE RETURN (0x000D)

- LINE FEED (0x000A)

- SPACE (0x0020)

In addition, these characters are ignored if the preceding character is a white-space character. The preceding character can be contained in the same element, in the parent element, or in the preceding sibling element, as long as it is contained within the same paragraph element and the element in which it is contained processes white-space characters as described above. White-space characters at the start or end of the paragraph are ignored, regardless whether they are contained in the paragraph element itself, or in a child element in which white-space characters are collapsed as described above.

These white-space processing rules shall enable authors to use white-space characters to improve the readability of the XML source of an OpenDocument document in the same way as they can use them in HTML.

§ 5.1.1, "White-space Characters", following sub-section "Line Breaks", p. 86

{The following text inserts a new sub-section, including reproduction of a new pattern definition "paragraph-content" (lines 1266-1268 of the amended schema) inserted into the normative Relax-NG schema.}

Soft Page Break

The `<text:soft-page-break>` element represents a soft page break within a heading or paragraph.

See section 2.3.1:Use Soft Page Breaks for details regarding soft page breaks.

```
1266    <define name="paragraph-content" combine="choice">
1267        <ref name="text-soft-page-break"/>
1268    </define>
```

§ 5.1.4, "Hyperlinks", paragraph 3, p. 87

The attributes that may be associated with the `<text:a>` element are:

- Name
- Title
- Link location
- Target frame
- Text styles

§ 5.1.4, "Hyperlinks", following sub-section "Name", p. 87

{The following text inserts a new sub-section, including reproduction of a new pattern definition 'text-a-attlist' (lines 1304-1310 of the amended schema) inserted into the normative Relax-NG schema.}

Title

The `office:title` attribute specifies a short accessible description for hint text. See appendix E for guidelines how to use this attribute.

```
1304    <define name="text-a-attlist" combine="interleave">
1305        <optional>
1306            <attribute name="office:title">
1307                <ref name="string"/>
1308            </attribute>
1309        </optional>
1310    </define>
```

§ 6.4.14, "Document Modification Time", paragraph 2, p. 120

This element displays the information from the `<dcmeta:date>` element. The name was chosen to avoid confusion with `<text:date>` fields.

§ 6.4.15, "Document Modification Date", paragraph 2, p. 121

This element displays the information from the `<dcmeta:date>` element. The name was chosen to avoid confusion with `<text:date>` fields.

§ 6.6.9, "DDE Connection Fields", following paragraph 2, p. 137

{The following text inserts a new paragraph following paragraph 2.}

See section 12.6 for the use of DDE connections.

§ 6.7.6, "Formula", paragraph 2, p. 143

The formula should start with a namespace prefix that indicates the syntax and semantic used within the formula.

§ 8.1, "Basic Table Model", paragraph 3, p. 178

{The following text creates a new paragraph by inserting a paragraph break in the existing paragraph.}

Table rows may be empty, and different rows might contain a different number of table cells. This is not an error, but applications might resolve this in different ways. Spreadsheet applications typically operate on large tables that have a fixed application dependent row and column number, but may have an unused area. Only the used area of the table is saved in files. When loading a table with empty or incomplete rows into a spreadsheet application, empty rows typically introduce a default row (just as in an empty sheet), and incomplete rows are filled with empty cells (just like in an empty sheet).

All other applications typically have fixed size tables. Incomplete rows are basically rendered as if they had the necessary number of empty cells, and the same applies to empty rows. Empty cells typically occupy the space of an empty paragraph.

§ 8.1.1, "Table Element", schema fragment preceding sub-section "Table Name", p. 180

{The following text inserts three new lines in the definition of the pattern "table-rows" (lines 3579-3581 of the amended schema) into the normative Relax-NG schema as reproduced in this section.}

```
3575  <define name="table-rows">
3576      <choice>
3577          <ref name="table-table-rows"/>
3578          <oneOrMore>
3579              <optional>
3580                  <ref name="text-soft-page-break"/>
3581              </optional>
3582              <ref name="table-table-row"/>
3583          </oneOrMore>
3584      </choice>
3585  </define>
```

§ 8.1.1, "Table Element", following sub-section "Print Ranges", p. 181

{The following text inserts a new sub-section.}

Soft Page Breaks

The `<text:soft-page-break>` element represents a soft page break between two table rows. It may appear in front of `<table:table-row>` elements.

See section 2.3.1:Use Soft Page Breaks for details regarding soft page breaks.

§ 8.1.2, "Table Row", sub-section "Default Cell Style", paragraph 1, p. 182

The `table:default-cell-style-name` attribute specifies a default cell style. Cells contained in the row that don't have a `table:style-name` attribute use this~~without an individual cell style use these~~ default cell style.

§ 8.1.2, "Table Row", sub-section "Default Cell Style", following paragraph 1, p. 182

{The following text inserts a new paragraph.}

The attribute is applied to cells that are defined by a `<table:table-cell>` element. It is typically not applied to table cells that spreadsheet application may display in addition to those defined in the document.

§ 8.1.3, "Table Cell", paragraph 2, p. 184

The `<table:table-cell>` element is very similar to the table cell elements of [XSL] and [HTML4], and the rules regarding cells that span several columns or rows that exist in HTML and XSL apply to the OpenDocument specification as well. This means that there are no `<table:table-cell>` elements in the row/column grid for positions that are covered by a merged cell, that is, that are covered by a cell that spans several columns or rows. The `<table:covered-table-cell>` element exists to be able to specify cells for such positions . It has to appear wherever a position in the row/column grid is covered by a cell that spans several rows or columns. Its position in the grid is calculated by assuming a column and row span of 1 for all cells regardless whether they are specified by a `<table:table-cell>` or a `<table:covered-table-cell>` element. The `<table:covered-table-cell>` is especially used by spreadsheet applications, where it is a common use case that a covered cell contains content.

§ 8.1.3, "Table Cell", sub-section "Cell Content Validation", paragraph 1, p. 185

The `table:content-validation-name` attribute specifies if a cell contains a validity check. The value of this attribute is the name of a `<table:`~~cell-~~`content-validation>` element. If the attribute is not present, the cell may have arbitrary content.

§ 8.1.3, "Table Cell", sub-section "Cell Content Validation", paragraph 2, p. 186

See section 8.5.3 for more information on cell content validation and the `<table:`~~cell-~~`content-validation>` element.

§ 8.2.1, "Column Description", sub-section "Default Cell Style", paragraph 1, p. 190

The `table:default-cell-style-name` attribute specifies the default cell style. Cells that don't have a `table:style-name` attribute~~without a style~~ use this style when there is no default cell style specified for the cell's row as well.

§ 8.2.1, "Column Description", sub-section "Default Cell Style", following paragraph 1, p. 190

{The following text inserts a new paragraph.}

The attribute is applied to cells that are defined by a `<table:table-cell>` element. It is typically not applied to table cells that spreadsheet application may display in addition to those defined in the document.

§ 8.2.2, "Header Columns", preceding paragraph 1, p. 190

{The following text inserts a new paragraph at the start of the section.}

For accessibility purposes, header information is needed. Therefore, any columns designated as headers by the author must be tagged as such by encapsulating them within a `<table:table-header-columns>` element. Using style information only to designate header columns is insufficient.

§ 8.2.2, "Header Columns", paragraph 1, p. 190

If a table does not fit on a single page, ~~a set of adjacent table columns can be automatically repeated on every page. To do so, their columns descriptions have to be included in a `<table:table-header-columns>` element. Descriptions of columns that shall not be repeated on every page can be included into a `<table:table-columns>` element, but don't have to.~~ table columns that are included in a `<table:table-header-columns>` element are automatically repeated on every page. A table must not contain more than one `<table:table-header-columns>` element, and a `<table:table-columns>` must not follow another `<table:table-columns>` element~~.~~ ~~T.~~ with the only exception ~~are~~of tables that contain grouped columns (see 8.2.3). Such tables may contain more than one `<table:table-header-columns>` element, provided that they are contained in different column groups and the columns contained in the elements are adjacent.

§ 8.2.4, "Header Rows", preceding paragraph 1, p. 191

{The following text inserts a new paragraph at the start of the section.}

For accessibility purposes, header information is needed. Therefore, any rows designated as headers by the author must be tagged as such by encapsulating them within a `<table:table-header-rows>` element. Using style information only to designate header rows is insufficient.

§ 8.2.4, "Header Rows", paragraph 1, p. 191

If a table does not fit on a single page, ~~a set of adjacent table rows can be automatically repeated on every page. To do so, their row elements have to be included in a `<table:table-headerrows>` element. Rows that shall not be repeated on every page can be included into a `<table:table-rows>` element, but don't have to. A table must not contain more than one `<table:table-header-rows>` element, and a `<table:table-rows>` must not follow another `<table:table-rows>` element. The only exception are tables~~table rows that are included in a `<table:table-header-rows>` element are automatically repeated on every page. A table must not contain more than one `<table:table-header-rows>` element. The one exception to this is a table that contain~~s~~ grouped rows (see 8.2.5). Such a table~~s~~ may contain more than one `<table:table-header-rows>` element, provided that they are contained in different row groups and the rows contained in the elements are adjacent.