

PERSONALIZED PRINT



MARKUP LANGUAGE

PODi

PPML Imposition Version 3.0

May 2010

Personalized Print Markup Language Imposition Specification

PODi: the Digital Printing Initiative
1240 Jefferson Road, Rochester, New York 14623, USA
Tel: (585) 239-6014
Internet: <http://www.podi.org>



PODi the Digital Printing Initiative

Approval of a PODi standard requires acceptance by the members of PODi.

PODi is a not for profit industry consortium formed in 1996. Its charter is to foster the growth of the digital printing industry through market and standards development activities. PODi constantly monitors market and technology trends in the industry, and shares information through seminars, independent research, white papers, articles, and the web. PODi promotes interoperability through the PPML suite of open, XML based standards, test suites and certification.

PODi welcomes feedback on this specification, and offers the following services to support widespread adoption of the specification:

Specification Updates

The PPML Functional specification is distributed free of charge. Developers who are implementing the PPML standard are invited to subscribe to free PPML updates and the technical note service.

Developer Support web site

Software and hardware developers interested in supporting PPML are invited to register for the PPML Developers discussion group.

To participate in the PPML initiative, send an email to ppmlinfo@podi.org.

© 2010 PODi: the Digital Printing initiative, All rights reserved

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise without the prior written permission of the publisher

Contents

PODi the Digital Printing Initiative	ii
Contents	iii
Foreword	iv
Personalized Print Markup Language Impositioning Specification	1
1 Scope	1
2 Normative references	1
3 Conformance.....	1
4 Technical requirements	2
4.1 Data structure	3
5 Syntax and semantics.....	5
5.1 The <PRINT_LAYOUT> Element	5
5.2 The <PAGE_LAYOUT> Element.....	7
5.3 The <SHEET_LAYOUT> Element	10
5.4 The <SHEET_MARK> Element	11
5.5 The <IMPOSITION> Element.....	12
5.6 The <IMPOSITION_REF> Element	14
5.7 The <SIGNATURE> Element.....	16
5.8 The <CELL> Element.....	18
5.9 The <HOR_TRIM_MARKS> Element.....	24
5.10 The <VER_TRIM_MARKS> Element	26
5.11 The <HOR_GUTTER> Element	27
5.12 The <VER_GUTTER> Element.....	29
5.13 The <HOR_FOLD_MARKS> Element.....	30
5.14 The <VER_FOLD_MARKS> Element	31
5.15 The <REPEAT> Element	32
Annex A (informative) Revision history	36
A.1 Version 3.0 May 2010	36
Index	37

Foreword

PODi: the Digital Printing Initiative developed this Imposition Specification to identify the specific requirements for how to use Imposition functions in a PPML environment.

This specification is intended to be a companion to the PPML Functional Specification. Readers are directed to the PPML Functional specification for the definition of the syntax and semantics of PPML.

PODi does not guarantee the suitability of any of PPML or any of the conformance subsets for any specific purpose.

The working group responsible for the current specification had the following membership:

PODi Senior Technologist: Dr. Paul Jones

PODi Director of Technology: James Mekis

Contributing working group members:

EFI: Boris Aronshtam, Reuven Ackner

Hewlett-Packard: Steve Hiebert

IBM: Hitesh Bhindi, Claudia Alimpich, Art Ford

Kodak Creo: Luci Wahrmann

Kodak NexPress: Tim Donahue

Kodak Versamark: Josh Howard, Pat McGrew

Konica Minolta: Darrell Hopp

Océ Printing Systems: Helmut Weiner

Pageflex: Peter Davis

Punch Graphix: Bart Wynants

Xerox: John Czudak

Send suggestions for improving this standard to PODi, 1240 Jefferson Road, Rochester, NY 14623, USA; e-mail: ppmlinfo@podii.org.

Personalized Print Markup Language Impositioning Specification

1 Scope

This specification identifies the data formats and semantics for implementing impositioning using Personalized Print Markup Language. For information on the use of PPML with the CIP4 Job Definition Format (JDF) see the *PODi Digital Print Specification*.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including amendments) applies.

PODi: the Digital Printing Initiative. *Personalized Digital Print Ticket* (all versions, 1.5/2.1/2.2).

International Cooperation for the Integration of Processes in Prepress, Press and Postpress (CIP4): System Behaviour and Interoperability WG. December 22, 2004. *Base ICS, Version: 1.0*.

Adobe Systems. *Postscript Language Reference Manual* (2nd Edition). December 1990. Addison-Wesley Professional.

PPML is specified both in a Document Type Definition (DTD, <http://www.w3.org/TR/1998/REC-xml-19980210#dt-doctype>) and in an XML Schema (<http://www.w3.org/XML/Schema>). All versions of the DTD and Schema will always be available at <http://www.podi.org/ppml>.

3 Conformance

A conforming PPML dataset shall adhere to the technical requirements set forth in the *Personalized Print Markup Language Functional Specification*.

4 Technical requirements

The PPML language includes **imposition**. It is important to understand what imposition is and is not, especially in the context of personalized documents, which are a main purpose of the PPML language.

- **Imposition** is the process of positioning page images on sheets of paper in the printer (or in a digital printing press), as part of the process of producing finished documents.
- In addition to the page images, various marks can be added to the sheets, to aid in the production process. For instance, marks can be added to show where the paper should be folded or trimmed.
- Imposition has no effect on the content of any individual page – it only affects where the pages are placed on a press sheet.

In this document, “imposition” (lowercase) refers to the functions described above. It does not refer to processing of the **IMPOSITION** element. “Imposing Consumers” are ones that process the **SHEET LAYOUT** element.

NOTE PPML Consumers are not required to support the **SHEET_LAYOUT** element, nor the *Ncopies* and *Collate* attributes on **PRINT_LAYOUT**. This means a complex production job intended for a large-format digital printing press can be proof-printed on a simpler, small-format desktop printer. Similarly, a single-page production printer can print the dataset’s document content stream (including copies and collation), ignoring imposition instructions.

It also means a post-processing system can extract the document content stream (Document Sets, Documents and Pages) from a PPML dataset, and use other methods to assign pages to sheets, add sheet marks, etc.

In non-personalized printing, imposition is the placement of unchanging master pages onto a reproduction master, such as a printing plate.

However, in digital printing of personalized documents, every copy is unique. Therefore, in addition to the regular imposition instructions, the language must also specify where to place each sequential copy of the document (each Instance Document). Sometimes the next document starts on a separate sheet, sometimes it starts in the next row of the same sheet, and sometimes it starts in the next column of the same sheet.

4.1 Data structure

This section provides a conceptual overview of how PPML pages are printed onto sheets as part of the overall production process. Each element is defined in its own section below.

4.1.1 Top level elements

<PRINT_LAYOUT> *includes:*
 <PAGE_LAYOUT> defines page size and cropping.
 <SHEET_LAYOUT> defines the size of the sheet,
 the sheet marks (e.g. crop marks),
 and all imposition instructions

4.1.2 Sheet layout elements

Sheet layout elements include imposition elements plus certain production marks that are associated with each sheet.

<SHEET_LAYOUT> *includes:*
 <SHEET_MARK>
 <IMPOSITION> elements

4.1.3 Imposition elements

Imposition elements contain signature definitions and **REPEAT** elements:

<IMPOSITION> *includes:*
 <SIGNATURE> *or*
 <REPEAT>

Impositions can have a *Name* and can be referenced with **IMPOSITION_REF**.

4.1.4 Signature elements

Finally, the **SIGNATURE** element (and its surrounding **REPEAT**s, if any) define what is to be printed on a single sheet:

<SIGNATURE>
 <CELL> defines the page order of each available
 position in the imposition layout:
 which location should receive the first Page,
 the second Page, and so on,
 and whether the Page should be rotated.
 Gutter locations & sizes (spaces between cells)
 Fold marks
 Trim marks

4.1.5 Production marks

“Production marks” are marks added to the sheet to assist in production; they are not part of document content. The Consumer may add production marks to a sheet after all the pages have been imaged, or before the pages, or both. Production marks and **PAGEs** are imaged in the order they appear.

The following sections define each of the elements presented in this overview.

5 Syntax and semantics

5.1 The <PRINT_LAYOUT> Element

5.1.1 Model

PRINT_LAYOUT (PAGE_LAYOUT, SHEET_LAYOUT?)

NOTE The use of this element is defined in the model(s) of the following element(s): **PPML** and **DOCUMENT_SET** or **JOB**.

5.1.2 Attributes

Attribute	Required /Optional	Type	Description
Ncopies	Optional	Integer	How many copies to print of each sheet (for an imposing Consumer) or each page (for a non-imposing Consumer). Default=1.
Collate	Optional	Keyword	"Document" (default) = print the entire first copy of the document (all sheets, all pages), then print the entire second copy of the same document, and subsequent copies. "DocSet" = print one copy of the entire Document Set (one copy of each document), then print the entire set again (another copy of each document), and subsequent copies. "No" = print all copies of the first sheet (or page) of the first document, then print all copies of the second sheet (or page) of that document, and subsequent copies.

Notice the distinction between the non-imposing consumer (which sees only pages) and the imposing consumer (which sees sheets). Both *Ncopies* and *Collate* make sense for both environments. The non-imposing consumer will copy and collate individual pages, while an imposing consumer will copy and collate full sheets.

Generally, uncollated output makes fewer demands on the Consumer's memory and may thus be the preferred mode when outputting to lower-powered products.

5.1.3 Description

PRINT_LAYOUT is the master element that includes the page dimensions and how the Pages are to be laid out onto sheets by the Consumer.

5.1.3.1 Example

The following illustrates a simple setup for printing letter-size pages onto 12x18" sheets. (Lower-level elements are omitted for this illustration.)

```
<PRINT_LAYOUT>
  <PAGE_LAYOUT TrimBox="0 0 612 792"/>
  <SHEET_LAYOUT HSize="1296" VSize="864">
    ...
  </SHEET_LAYOUT>
</PRINT_LAYOUT>
```

5.2 The <PAGE_LAYOUT> Element

5.2.1 Model

PAGE_LAYOUT EMPTY

NOTE The use of this element is defined in the model(s) of the following element(s): **PRINT_LAYOUT** and **SHEET_LAYOUT**.

5.2.2 Attributes

Attribute	Required /Optional	Type	Description
TrimBox	Required	Number × 4	Coordinates, in 1/72", of the trimmed size of the final page (i.e. after finishing).
BleedBox	Optional	Number × 4	Coordinates, in 1/72", of the page's bleed area. Defaults to TrimBox.
BoundingBox	Optional	Number × 4	Coordinates, in 1/72", of the maximum area that may need to be printed. Defaults to BleedBox.

5.2.3 Description

The **PAGE_LAYOUT** element describes page cropping information when using PPML's imposition. This element appears similar to **PAGE_DESIGN** because both have a *TrimBox* and *BleedBox* attribute.

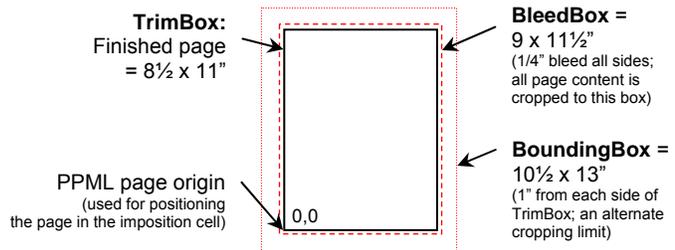
The **PAGE_LAYOUT** element states the rectangular dimensions of the Page. Three different dimensions are given: the trim box, the bleed box, and the bounding box.

Example:

```
<PAGE_LAYOUT TrimBox="0 0 612 792"
  BleedBox="-18 -18 630 810"
  BoundingBox="-72 -72 684 864" />
```

5.2.3.1 The Trim Box

TrimBox indicates the final page size after trimming. The lower left corner of the trimmed page is the origin: when *BleedBox* or *BoundingBox* extends outside the trimmed page, its lower left corner will have negative coordinates, as shown in the **PAGE_LAYOUT** example above.



5.2.3.2 The Bleed Box

Bleed is the practice of intentionally allowing page content to extend a small distance beyond TrimBox. This is done to compensate for normal imperfections in the finishing process: if the trimming is not perfectly accurate, blank paper might be visible along the edge of the page. Extending the page image beyond TrimBox (i.e. using bleeds) avoids this.

The **PAGE_LAYOUT** element's *BleedBox* attribute specifies how far the image area is *allowed* to extend outside the page, but the allowed amount may not always be used. For instance, at the edge of a sheet, the entire specified bleed area is used. But within an imposed sheet (i.e. between two adjacent pages), the bleed extends into the gutter between the pages (if there is one) as follows:

- If there is no space (gutter) between the pages, then no bleed is needed at that edge. On that side, the Consumer crops the content of each page at each Page's TrimBox (which in this case is also the line where the two pages meet). At the outside borders of the signature the bleed would still be used.
- If there is a gutter, and it's less than or equal to the bleed, then the bleed fills the gutter. The bleed from each page stops in the middle of the gutter.
- If the gutter is wider than the bleed, the Consumer crops the page image at the BleedBox.

If no BleedBox is specified, BleedBox defaults to TrimBox.

TrimBox should not extend beyond BleedBox, but if it does, TrimBox will prevail.

5.2.3.3 The Bounding Box

The bounding box states the farthest uncropped extent of all objects on the page. In rare circumstances this may be useful as an alternative cropping boundary. It is expected to be used less frequently than *BleedBox* but will be of value in appropriate applications.

For instance, a PPML Page could consist of a single full-page object created by a desktop publishing application. Output from such applications typically includes production marks that fall outside the page area: crop marks, file identification information, etc. When the PPML page containing this object is imposed, the Producer has typically set *BleedBox* to a small value, so that all the application's production mark information is cropped out.

When the same PPML page is proof-printed on a non-imposing printer, it may be preferable not to crop out those marks. With *BoundingBox*, a Producer can indicate the farthest uncropped extent of all objects on the page. The Consumer can honor *BoundingBox* instead of *BleedBox*, which allows printing page proofs that show the original application-provided marks outside the bleed area.

If *BoundingBox* is not specified, it defaults to *BleedBox*. If *BleedBox* extends beyond *BoundingBox*, then *BoundingBox* is set to the intersection of the two.

5.2.4 Page orientation

All dimensions in the attributes are to be listed in "upright" orientation. For instance, a portrait letter-size page will have `TrimBox="0 0 612 792"` and a landscape letter-size page will have `TrimBox="0 0 792 612"`. Thus, no separate *Orientation* attribute is needed.

Any Page may be rotated when it is used in **IMPOSITION** and/or **SHEET_LAYOUT**. But the Page itself, and its content, are independent of imposition and printing.

5.3 The <SHEET_LAYOUT> Element

5.3.1 Model

SHEET_LAYOUT (SHEET_MARK | (PAGE_LAYOUT?, (IMPOSITION | IMPOSITION_REF))) *

NOTE The use of this element is defined in the model of the following element: **PRINT_LAYOUT**.

5.3.2 Attributes

Attribute	Required /Optional	Type	Description
Hsize	Required	Number	Horizontal size of the sheet in 1/72"
Vsize	Required	Number	Vertical size of the sheet in 1/72"
GangDocuments	Optional	Boolean	Yes means all Instance Documents in a Document Set are to be concatenated into a single stream of pages for imposition. No (the default) means each Instance Document must start a new sheet. See also the PageOrder attribute of the CELL element (section 0, "7.8.5 Using expressions in the PageOrder attribute").

5.3.3 Description

In general, the **SHEET_LAYOUT** element contains all the elements that define what goes where on which sheet. It declares any marks that are associated with the sheet itself and what imposition instructions to use.

5.3.4 Semantics

The model allows **SHEET_MARK** elements to come before or after imposition, or before *and* after Imposition elements. The Consumer must image the sheet in the sequence specified in **SHEET_LAYOUT**.

If **SHEET_LAYOUT** contains no child elements, then it defines nothing but the sheet size – it defines no imposition or sheet marks. In this case each page is centered on a sheet. If the **PAGE**'s BleedBox is bigger than the sheet size, then the sheet size is used for cropping.

An optional **PAGE_LAYOUT** element may precede **IMPOSITION** or **IMPOSITION_REF**, in which case it replaces the previous **PAGE_LAYOUT**. This allows combining several different imposition schemes on the same sheet, including (optionally) different page sizes.

One set of page numbers applies to the whole sheet, even if it contains more than one **IMPOSITION**.

5.4 The <SHEET_MARK> Element

5.4.1 Model

SHEET_MARK (OCCURRENCE_REF)

NOTE The use of this element is defined in the model of the following element: SHEET_LAYOUT.

5.4.2 Attributes

Attribute	Required /Optional	Type	Description
Position	Required	Number × 2	Location where the bottom left corner of the mark's bounding box is to be placed on the sheet.
Face	Optional	Keyword	Whether the Sheet Mark is to appear on the top of the sheet (Face= "Up") or bottom of the sheet (Face= "Dn"). Default=Up.

5.4.3 Description

The SHEET_MARK element places a Reusable Object at a specified location on every sheet. Applications of this feature are expected to include color control strips, the print shop's logo, or job ID information.

A sheet mark may be placed anywhere on the sheet: the Producer may place sheet marks on top of page image areas if desired. The name of the Occurrence Reference is resolved immediately when the Sheet Mark element is encountered. That is, the OCCURRENCE content object named in the OCCURRENCE_REF element is retrieved immediately, such that even if the OCCURRENCE is renamed while the job is running, the appearance of the SHEET_MARK will not be affected.

This element can only exist at the PPML or Document Set level (not Document or Page) because its enclosing SHEET_LAYOUT element can only appear at those levels. Therefore, the Occurrence used in a Sheet Mark cannot have a scope of Document or Page.

5.4.4 Future considerations: variable sheet marks

Future versions may include the ability to imprint variable information in a sheet mark. Examples might include the date and time of the press run, a text string to identify which machine printed the sheets, a sheet number within the run or within the job, and so on.

5.5 The <IMPOSITION> Element

5.5.1 Model

IMPOSITION (SIGNATURE | REPEAT)

NOTE The use of this element is defined in the models of the following elements: **SHEET_LAYOUT**, **DOCUMENT_SET**, and **PPML**.

5.5.2 Attributes

Attribute	Required /Optional	Type	Description
Name	Optional	String	Optional identifying string for reference in a subsequent IMPOSITION_REF element.
Environment	Required if Scope="Global"; not needed otherwise	String	Specifies the environment in which the Imposition should be defined. (There is no default environment.)
Scope	Optional	Keyword	Specifies the scope of this Imposition template's use. Valid values for <i>Scope</i> are Global, PPML, and DocSet. By default, the scope is the containing element in which the imposition is defined. A higher value may be specified in this attribute, but a lower value is an error.
Rotation	Optional	Integer	Rotation of the IMPOSITION content structure (the imposed set of signatures) relative to the sheet, counterclockwise: 0, 90, 180, 270 degrees. Default=0.
Position	Optional	Number × 2	Location where the bottom left corner of the rotated IMPOSITION content structure is to be placed on the sheet. If the Position attribute is not used, the entire structure is centered on the sheet. The imposition content structure is the logical structure that contains all the cells (including any empty cells) in the imposition scheme. It does not include any trim or fold marks.

5.5.3 Description

The **IMPOSITION** element creates an imposition template, which immediately becomes the active imposition. The optional `Name` attribute allows saving it as a reusable template so it can be recalled with **IMPOSITION_REF**.

The **IMPOSITION** element can have two possible content structures:

For multi-sheet applications, **IMPOSITION** can contain a **SIGNATURE**, e.g.:

```
<IMPOSITION>
  <SIGNATURE> ... </SIGNATURE>
</IMPOSITION>
```

For applications where the document is smaller than one sheet, the **IMPOSITION** element can contain one **REPEAT** element (which may be nested) around one **SIGNATURE** element, for instance:

```
<IMPOSITION>
  <REPEAT Direction="Stack">
    <REPEAT Direction="Hor">
      <REPEAT Direction="Ver">
        <SIGNATURE> ... </SIGNATURE>
      </REPEAT>
    </REPEAT>
  </REPEAT>
</IMPOSITION>
```

5.6 The <IMPOSITION_REF> Element

5.6.1 Model

IMPOSITION_REF EMPTY

NOTE The use of this element is defined in the model of the following element: **SHEET_LAYOUT**.

5.6.2 Attributes

Attribute	Required /Optional	Type	Description
Name	Required	String	Name attribute of the imposition template previously defined.
Environment	Optional	String	The environment in which the name of a Global imposition template should be interpreted. This attribute is required if the scope of the template is Global; otherwise, this attribute has no meaning.
Rotation	Optional	Number	Rotation of the IMPOSITION content structure, counterclockwise: 0, 90, 180, 270 degrees. Default=0.
Position	Optional	Number × 2	Location where the bottom left corner of the IMPOSITION content structure is to be placed on the sheet. If the Position attribute is not used, the entire structure is centered on the sheet.

5.6.3 Description

The **IMPOSITION_REF** element recalls an imposition template that was previously defined. This enables the convenience of creating a library of standard imposition setups and reusing them.

5.6.4 Implementation notes

Calling for a stored imposition template by name has advantages but also has a side effect. Producers should be conscious of this.

One advantage is that the dataset can be marginally smaller. Another is that it may be simpler for the Producer to output a simple name than to regenerate all the imposition instructions. If a dataset uses **IMPOSITION_REF** to call for a template by name, then the latest version of that template will automatically be retrieved. This means that if a shop has refined its template, the updates will automatically be implemented in any dataset that uses that template.

It also means that the dataset no longer has complete control of the imposition: by definition, **IMPOSITION_REF** means “I don’t care what imposition is stored under this name – use it.”

If the Producer requires absolute control of the imposition for a job, it should explicitly define the imposition in the dataset, using **IMPOSITION** and its child elements. The dataset can still use **IMPOSITION_REF** to call the imposition by name later in the dataset; the point is that the imposition is only certain if it's defined within the dataset that references it.

5.7 The <SIGNATURE> Element

5.7.1 Model

SIGNATURE (CELL+, HOR_TRIM_MARKS?, VER_TRIM_MARKS?,
 HOR_GUTTER*, VER_GUTTER*,
 HOR_FOLD_MARKS*, VER_FOLD_MARKS*)

NOTE The use of this element is defined in the models of the following elements: **IMPOSITION** and **REPEAT**.

5.7.2 Attributes

Attribute	Required /Optional	Type	Description
Nrows	Required	Integer	The number of rows in this signature.
Ncols	Required	Integer	The number of columns in this signature.
PageCount	Optional	Integer	The number of <i>different</i> pages consumed by this signature. Default is the number of CELL elements in this signature.

5.7.3 Description

A **signature** is a set of one or more pages from an Instance Document, printed on a single sheet of paper. The pages are arranged in a specific sequence, and are printed on one or both sides of the sheet.

The **SIGNATURE** element defines a uniform cell grid defined by *Nrows* and *Ncols*. The size of the cells in the grid is not specified by the imposition layout, but is defined by the *TrimBox* attribute of the **PAGE_LAYOUT** of the document that is imposed. The **HOR_GUTTER** and **VER_GUTTER** elements define the spacing between the cells in the grid.

Every cell has the same size. Specifically, the *Rotation* attribute of the **CELL** is not used to determine the size of a cell.

Once this grid is defined, the *BleedBox* in the **PAGE_LAYOUT** defines the clipping rectangle for each cell depending on the gutters and the relative position in the grid.

The *Rotation* and *Position* in the **IMPOSITION** element determine how and where this grid is positioned.

5.7.4 PageCount applications

PageCount specifically states how many different pages the Producer has assigned to this Signature. Typically this equals the number of **CELL** elements, but that is not required.

For instance, in an eight-page signature the Producer may choose to assign only four or six pages to the signature, and that's the number that would be assigned to PageCount. As another example, a Producer may want to assign the same page to multiple locations in the same signature – for instance it may duplicate the second page on the signature, for some reason. In that case when the Producer calculates PageCount, it would ignore those duplicates, counting only how many *different* pages are assigned to the signature.

5.8 The <CELL> Element

5.8.1 Model

CELL EMPTY

NOTE The use of this element is defined in the model of the following element: **SIGNATURE**.

5.8.2 Attributes

Attribute	Required /Optional	Type	Description
Row	Required	Integer	Row number of the cell being defined. Top row=1.
Col	Required	Integer	Column number of the cell being defined. Left column=1.
Face	Optional	Keyword	Whether the Page is to appear on the top of the sheet (Face="Up") or bottom of the sheet (Face="Dn"). Default=Up.
PageOrder	Required	String	Defines the sequence number of the Page to be placed in this cell. Can be an integer or an expression..
Rotation	Optional	Integer	Rotation of the Page, counterclockwise: 0, 90, 180, 270. Default=0.

5.8.3 Description

The **CELL** element assigns Pages to specific locations on a Signature. For each Page, it specifies the row and column position within the signature, whether the Page is to be printed on the face-up or face-down side of the sheet, and whether the page content is to be rotated in the cell.

One **CELL** element may be used for each page position on either side of the signature. No **CELL** element has to be specified for positions that are empty.

The *TrimBox* attribute of the **PAGE_LAYOUT** used to instantiate the **IMPOSITION** template determines the actual size of every Cell within the Signature.

The *Rotation* attribute of the **CELL** determines how the page content is placed inside the Cell. It does not affect its size or bleed area. E.g. if the Rotation is 90, the page content is rotated 90 degrees counterclockwise around the center of the cell.

No trim marks will be generated for missing cells.

5.8.4 Using expressions in the *PageOrder* attribute

Expressions can use the operators +, −, *, /, and parentheses, operating on integers and two variables: *s* for sheet number (starting at 1) and *n* for number of pages to be imposed. Expressions are evaluated with normal operator precedence. Multiplication must be expressed by explicitly including the * operator – that is, use “2*s”, not “2s”. Remainders are discarded.

For print applications where page count varies from Instance Document to Instance Document, PPML imposition templates can automatically assign pages to the correct Signature and Cell position. To use this feature, the Producer should specify the *PageOrder* attribute using expressions based on *n*.

The variable *n* depends on *p*, the total number of pages that need to be imposed. This number *p* on its turn depends on the value of the *GangDocuments* attribute of **SHEET_LAYOUT**:

GangDocuments="No" means each Instance Document must start on a new Sheet. In this case, *p* refers to the number of pages in the current Instance Document, and the Consumer will evaluate the *PageOrder* expression separately for each Instance Document.

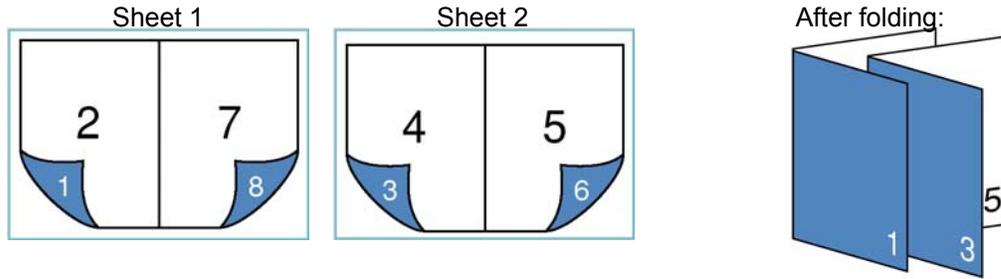
GangDocuments="Yes" means all Instance Documents are to be concatenated into a single stream of pages for imposition. In this case, *p* refers to the total page count (the sum of page counts for all documents in the Document Set) and *n* refers to a page's position in the concatenated stream of pages, not its position within its parent Document.

In both cases, *n* is derived from *p* according to the following rule: let *c* be the sum of all the *PageCount* attributes of all the **SIGNATURE** elements in the **SHEET_LAYOUT**, then *n* is *p* rounded up to the nearest multiple of *c*. The number of signatures generated will be *n/c*.

Any cell that has a resulting *PageOrder* attribute greater than *p* or less than 1 is left blank. For instance, if *c* equals 4, and an Instance Document contains 7 pages, then *n* for that document is 8, and the last cell will have no content.

5.8.5 Examples

This example shows an eight-page job being assigned to the cells of two four-page signatures.



The cell assignments shown in the above diagram for the eight pages are as follows. Pages that get assigned to the second signature are shown center-aligned so they're easy to recognize; notice that within each signature, the page sequence (as shown in the illustrations) is Down Up Up Down.

Page #	Signature	Row	Column	Face
1	1	1	1	Down
2	1	1	1	Up
3	2	1	1	Down
4	2	1	1	Up
5	2	1	2	Up
6	2	1	2	Down
7	1	1	2	Up
8	1	1	2	Down

This two-signature imposition can be described in a more general form, so that it handles any number of pages, and will automatically generate additional signatures as needed to accommodate those pages. This is done by using *one* 4-cell **SIGNATURE** element, with each *PageOrder* attribute being an expression *f* of *s*, the sheet number in the above table. The general form will be as follows.

NOTE “*f_n(s)*” is not part of the PPML code – it is explained below.

```
<IMPOSITION Name="2 x 2-UP Bundled">
  <SIGNATURE Nrows="1", Ncols="2">
    <CELL Row="1" Col="1" PageOrder= f1(s) Face="Up" Rotate="0"/>
    <CELL Row="1" Col="1" PageOrder= f2(s) Face="Dn" Rotate="0"/>
    <CELL Row="1" Col="2" PageOrder= f3(s) Face="Up" Rotate="0"/>
    <CELL Row="1" Col="2" PageOrder= f4(s) Face="Dn" Rotate="0"/>
  </SIGNATURE>
</IMPOSITION>
```

Each of the expressions *f₁(s) ... f₄(s)* is of the form *f_n(s)=a*s -b*.

The following illustrates how to determine $f_2(s)$, the expression for the second **CELL** element.

We find the values of a and b by rewriting the expression “ $a*s - b$ ” for the two entries in the above table that have Row=1, Col=1 and Face=Down (the first and third entry from the table). We know the result must be the page number shown in the first column of the table:

$$1 = a*1 - b \quad (\text{first entry from table})$$

$$3 = a*2 - b \quad (\text{third entry from table})$$

Resolving this for a and b gives $a=2$ and $b=1$. So the second **CELL** element becomes:

```
<CELL Row="1" Col="1" PageOrder="2*s-1" Face="Dn" Rotate="0"/>
```

Doing this for all four cells, the final code is:

```
<IMPOSITION Name="2 x 2-UP Bundled">
  <SIGNATURE Nrows="1", Ncols="2">
    <CELL Row="1" Col="1" PageOrder="2*s" Face="Up" Rotate="0"/>
    <CELL Row="1" Col="1" PageOrder="2*s-1" Face="Dn" Rotate="0"/>
    <CELL Row="1" Col="2" PageOrder="9-2*s" Face="Up" Rotate="0"/>
    <CELL Row="1" Col="2" PageOrder="10-2*s" Face="Dn" Rotate="0"/>
  </SIGNATURE>
</IMPOSITION>
```

The true power of using expressions in the *PageOrder* attribute is shown by generalizing the above for any n-page document, n being a multiple of 4:

```
<IMPOSITION Name="2 x 2-UP Bundled">
  <SIGNATURE Nrows="1", Ncols="2">
    <CELL Row="1" Col="1" PageOrder="2*s" Face="Up" Rotate="0"/>
    <CELL Row="1" Col="1" PageOrder="2*s-1" Face="Dn" Rotate="0"/>
    <CELL Row="1" Col="2" PageOrder="n+1-2*s" Face="Up" Rotate="0"/>
    <CELL Row="1" Col="2" PageOrder="n+2-2*s" Face="Dn" Rotate="0"/>
  </SIGNATURE>
</IMPOSITION>
```

If we instead want to fold each sheet first, then gather them together, the page assignment scheme would follow the same generic sequence but it would allocate pages 1-4 to the first signature, and 5-8 to the second signature, as follows:

Page #	Signature	Row	Column	Face
1	1	1	1	Down
2	1	1	1	Up
3	1	1	2	Up
4	1	1	2	Down
5	2	1	1	Down
6	2	1	1	Up
7	2	1	2	Up
8	2	1	2	Down

The resulting imposition is:

```
<IMPOSITION Name="2 x 2-UP">
  <SIGNATURE Nrows="1", Ncols="2">
    <CELL Row="1" Col="1" PageOrder="4*s-2" Face="Up" Rotate="0"/>
    <CELL Row="1" Col="1" PageOrder="4*s-3" Face="Dn" Rotate="0"/>
    <CELL Row="1" Col="2" PageOrder="4*s-1" Face="Up" Rotate="0"/>
    <CELL Row="1" Col="2" PageOrder="4*s-0" Face="Dn" Rotate="0"/>
  </SIGNATURE>
</IMPOSITION>
```

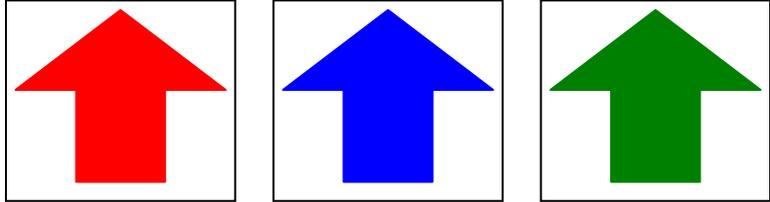
Notice that in this second example, every sheet is independent from the previous one, which is reflected by the *PageOrder* expressions being independent of *n*.

5.8.5.1 Cell Rotation Example

This example shows the effect of the *Rotation* attribute in a **CELL**. Some attributes have been omitted for clarity.

```
<SHEET_LAYOUT>
  <IMPOSITION>
    <SIGNATURE Nrows="2" Ncols="3">
      <CELL Row="1" Col="1" PageOrder="3*s" Rotation="0" />
      <CELL Row="2" Col="2" PageOrder="3*s+1" Rotation="270" />
      <CELL Row="1" Col="3" PageOrder="3*s+2" Rotation="180" />
    </SIGNATURE>
  </IMPOSITION>
</SHEET_LAYOUT>
```

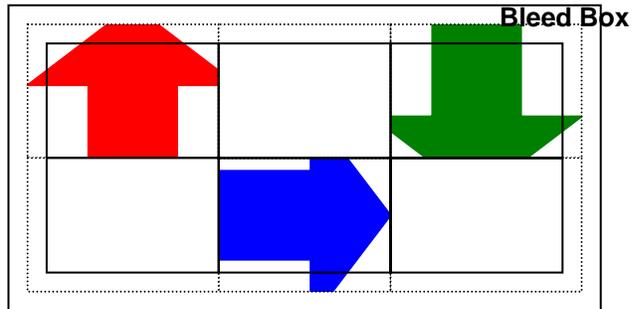
The input document contains these three pages :



The active **PAGE_LAYOUT** is as follows:



Executing the imposition gives the following result:



The trim and bleed boxes are shown in this drawing. They will not be visible in the actual PPML output.

5.9 The <HOR_TRIM_MARKS> Element

5.9.1 Model

HOR_TRIM_MARKS (OCCURRENCE_REF)

NOTE The use of this element is defined in the model of the following element: **SIGNATURE** .

5.9.2 Attributes

Attribute	Required /Optional	Type	Description
MarkDist	Optional	Number	Distance of the mark away from the page, in 1/72"
AllowOnPage	Optional	Boolean	Default= "No". If Yes, Trim Marks will <i>not</i> be suppressed if they fall on or inside another page's trim box.

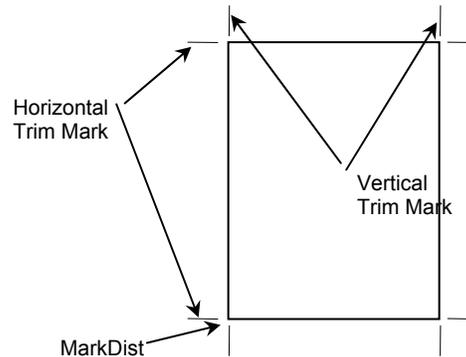
The following example shows how Trim Marks would be coded using two Reusable Object Occurrences named *VerTrim* and *HorTrim*. Each is to be positioned six points from the corner of its page.

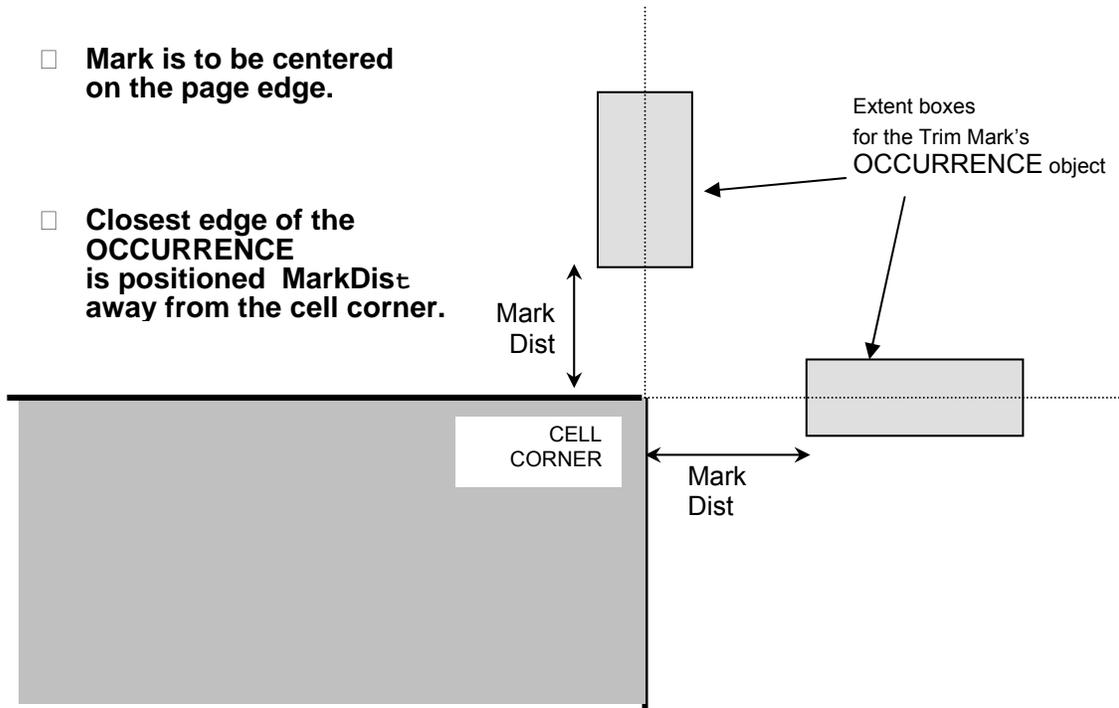
The Trim Marks elements are unaffected by how many cells are in the signature; they simply declare whether or not the signature has trim marks.

```
<IMPOSITION>
  <SIGNATURE Nrows="1" Ncols="2">
    <CELL .../>
    <CELL .../>
    <HOR_TRIM_MARKS MarkDist="6">
      <OCCURRENCE_REF Ref="HorTrim">
    </HOR_TRIM_MARKS>
    <VER_TRIM_MARKS MarkDist="6">
      <OCCURRENCE_REF Ref="VerTrim">
    </HOR_TRIM_MARKS>
  </SIGNATURE>
</IMPOSITION>
```

5.9.3 Description

Trim Marks are Reusable Object Occurrences that can be automatically placed by the Consumer on each sheet, at the corners of the final pages, on both sides of the sheet (front and back), using the **HOR_TRIM_MARKS** and **VER_TRIM_MARKS** elements.

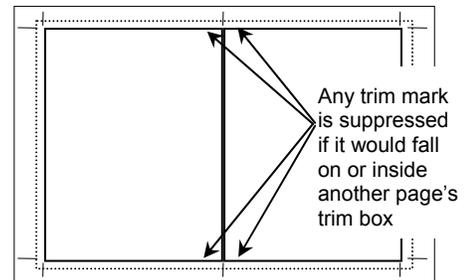




Each mark is centered on one boundary of the TrimBox. The attribute *MarkDist* specifies the mark's distance from the corner of the page.

The mark is printed without rotation or mirroring – for instance the vertical trim mark at the top of the page will be identical, relative to the sheet, to the mark at the bottom of the page.

If a signature has pages that touch, or nearly touch, as shown at right, some trim marks would fall onto the TrimBox of their neighboring pages. A trim mark is suppressed if any part of its bounding box falls closer than *MarkDist* to a neighboring trim box. An optional attribute *AllowOnPage="Yes"* (default = "No") can overrule this suppression.



The **OCCURRENCE_REF** may only refer to a reusable object with a scope at least as high as the scope of the **IMPOSITION** element enclosing this mark. It is an error to refer to a mark which is in scope, but which has a scope lower than that of the enclosing **IMPOSITION** element.

5.10 The <VER_TRIM_MARKS> Element

5.10.1. Model

VER_TRIM_MARKS (OCCURRENCE_REF)

NOTE The use of this element is defined in the model of the following element: **SIGNATURE** .

5.10.2 Attributes

Attribute	Required /Optional	Type	Description
MarkDist	Optional	Number	Distance of the mark away from the page, in 1/72"
AllowOnPage	Optional	Boolean	Default= "No". If Yes, Trim Marks will <i>not</i> be suppressed if they fall on or inside another page's trim box.

5.10.3 Description

The **VER_TRIM_MARKS** element is the vertical equivalent to **HOR_TRIM_MARKS**. See Section 5.9 for description, illustration, and examples.

The **OCCURRENCE_REF** may only refer to a reusable object with a scope at least as high as the scope of the **IMPOSITION** element enclosing this mark. It is an error to refer to a mark which is in scope, but which has a scope lower than that of the enclosing **IMPOSITION** element.

5.11 The <HOR_GUTTER> Element

5.11.1 Model

HOR_GUTTER EMPTY

NOTE The use of this element is defined in the model of the following element: **SIGNATURE**.

5.11.2 Attributes

Attribute	Required /Optional	Type	Description
Distance	Required	Number	Size (height) of the gutter, in 1/72"
BetweenRows	Required	Integer ×2	Identifies the set of rows between which this Distance applies. See examples. Top row = 1.

5.11.3 Description

The horizontal gutter is a horizontal strip of space between two rows of cells in a signature.

The *BetweenRows* attribute specifies the set of rows between which this gutter should be inserted. For instance, the following code shows a signature with *NRows*="3" and gutters between all rows:

```
<IMPOSITION>
  <SIGNATURE Nrows="3" Ncols="2">
    <CELL Row="1" Col="1" PageOrder="3" Face="Up" Rotate="180"/>
    ...
    <CELL Row="2" Col="2" PageOrder="8" Face="Dn" Rotate="0"/>
    <HOR_GUTTER BetweenRows="1 3" Distance="18"/>
  </SIGNATURE>
</IMPOSITION>
```

It is also possible to specify a different **HOR_GUTTER** element for each space between rows:

```
<IMPOSITION>
  <SIGNATURE Nrows="2" Ncols="2">
    <CELL Row="1" Col="1" PageOrder="3" Face="Up" Rotate="180"/>
    ...
    <CELL Row="2" Col="2" PageOrder="8" Face="Dn" Rotate="0"/>
    <HOR_GUTTER BetweenRows="1 2" Distance="36"/>
    <HOR_GUTTER BetweenRows="2 3" Distance="18"/>
  </SIGNATURE>
</IMPOSITION>
```

Each **HOR_GUTTER** element affects only the rows identified in *BetweenRows* – previous gutter settings for other gutters are unaffected. For instance, this code for an 8-row signature defines uniform spacing for all rows, then changes the value for the middle gutter to 1”:

```
<SIGNATURE Nrows="8" Ncols="1">
  ...
  <HOR_GUTTER BetweenRows="1 8" Distance="18"/>
  <HOR_GUTTER BetweenRows="4 5" Distance="72"/>
</SIGNATURE>
```

5.12 The <VER_GUTTER> Element

5.12.1 Model

VER_GUTTER EMPTY

NOTE The use of this element is defined in the model of the following element: **SIGNATURE**.

5.12.2 Attributes

Attribute	Required /Optional	Type	Description
Distance	Required	Number	Size (width) of the gutter, in 1/72"
BetweenCols	Required	Integer × 2	Identifies the set of columns between which this Distance applies. See examples. Columns are numbered from left to right.

5.12.3 Description

The **VER_GUTTER** element is identical to **HOR_GUTTER** except that it defines a vertical strip of space between two columns, not rows, of cells in a signature. See the description in section 5.11 for examples and explanation of attributes.

5.13 The <HOR_FOLD_MARKS> Element

5.13.1 Model

HOR_FOLD_MARKS (OCCURRENCE_REF)

NOTE The use of this element is defined in the model of the following element: **SIGNATURE** .

5.13.2 Attributes

Attribute	Required /Optional	Type	Description
BetweenRows	Required	Integer × 2	Rows between which the fold line exists
MarkDist	Optional	Number	Distance, in 1/72", between the outermost page of the signature and the start of the Reusable Object. (Positive value = away from the signature.)

5.13.3 Description

The **HOR_FOLD_MARKS** element specifies a pair of horizontal fold marks between two specified rows of a Signature – an Occurrence of a Reusable Object that will print outside the left and right edges of the Signature.

If fold marks are defined between two cells, the trim marks on the two corners of each cell closest to the fold marks are suppressed. Fold marks are also suppressed if any part of its bounding box falls closer than *MarkDist* from the trim box of a neighboring cell.

The name of the Reusable Object Occurrence is resolved immediately when this element is encountered. The **OCCURRENCE_REF** may only refer to a reusable object with a scope at least as high as the scope of the **IMPOSITION** element enclosing this mark. It is an error to refer to a mark which is in scope, but which has a scope lower than that of the enclosing **IMPOSITION** element.

5.13.4 Example

```
<HOR_FOLD_MARKS BetweenRows="1 2" MarkDist="6">
  <OCCURRENCE_REF Ref="HDashedLine"/>
</HOR_FOLD_MARKS>
```

5.14 The <VER_FOLD_MARKS> Element

5.14.1 Model

VER_FOLD_MARKS (OCCURRENCE_REF)

```
<VER_FOLD_MARKS BetweenCols="1 2" MarkDist="6">
  <OCCURRENCE_REF Ref="VFoldMark"/>
</VER_FOLD_MARKS>
```

NOTE The use of this element is defined in the model of the following element: **SIGNATURE** .

5.14.2 Attributes

Attribute	Required /Optional	Type	Description
BetweenCols	Required	Integer × 2	Columns between which the fold line exists
MarkDist	Optional	Number	Distance, in 1/72", between the outermost page of the signature and the start of the Reusable Object. (Positive value = away from the signature.)

5.14.3 Description

The **VER_FOLD_MARKS** element specifies a pair of vertical fold marks between two specified columns of a Signature – an Occurrence of a Reusable Object that will print outside the top and bottom edges of the Signature.

The mark will be centered on the fold line, at a distance specified by the *MarkDist* attribute.

If fold marks are defined between two cells, the trim marks on the two corners of each cell closest to the fold marks are suppressed. Fold marks are also suppressed if any part of its bounding box falls closer than *MarkDist* from the trim box of a neighboring cell.

The name of the Reusable Object Occurrence is resolved immediately when this element is encountered. The **OCCURRENCE_REF** may only refer to a reusable object with a scope at least as high as the scope of the **IMPOSITION** element enclosing this mark. It is an error to refer to a mark which is in scope, but which has a scope lower than that of the enclosing element.

5.15 The <REPEAT> Element

5.15.1 Model

REPEAT (REPEAT | SIGNATURE)

NOTE The use of this element is defined in the models of the following elements: **IMPOSITION** and **REPEAT**.

5.15.2 Attributes

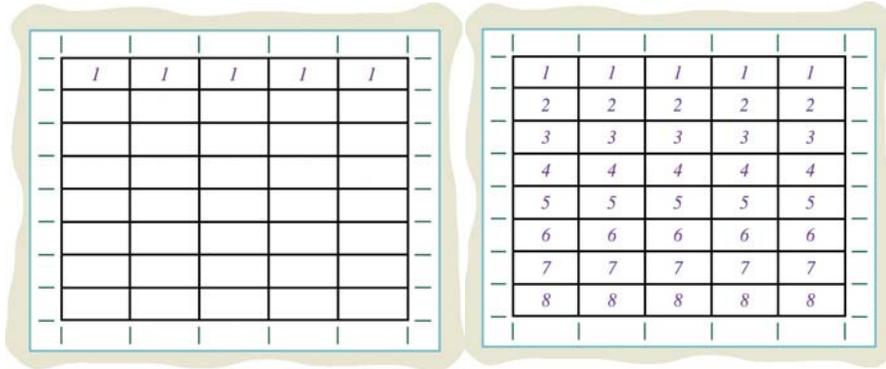
Attribute	Required /Optional	Type	Description
Direction	Required	Keyword	Specifies which direction this REPEAT element is defining. Allowable values: Ver (vertical), Hor (horizontal), Stack (from sheet to sheet).
Action	Required	Keyword	What to print in the next location: use the same value of the signature counter s again, or increment it. Values: Duplicate or Increment.
Order	Optional	Keyword	Values: Ascending (default) or Descending.
Count	Required	Integer	Total count of repeated instances.
Spacing	Optional	Number	Distance, in 1/72", between Signatures. Default=0. See the SpacingMethod attribute for the effect of this value.
SpacingMethod	Optional	Keyword	Values: Gap / Offset. Defines the meaning of the Spacing value: If Gap, then Spacing is the gap distance between signatures. If Offset, then Spacing is the distance from the start of one signature to the start of the next signature.

5.15.3 Description

An imposition template allows printing multiple pages on a signature, from one Instance Document. In contrast, the REPEAT element allows printing signatures from multiple documents on a single sheet. It also controls the distribution of different Instance Documents throughout the print run.

Three attributes control the effect of each REPEAT element: Direction, Action, and Count. The elements can be nested, with different values in each element. When REPEAT elements are nested, they are executed from innermost to outermost. For instance, the following code could be used in creating a sheet of five identical columns of eight different business cards (see illustration). (Inner elements have been omitted for this illustration.)

```
<REPEAT Direction="Ver" Action="Increment" Count="8">
  <REPEAT Direction="Hor" Action="Duplicate" Count="5">
    <SIGNATURE...>.... </SIGNATURE>
  </REPEAT>
</REPEAT>
```



1: The inner REPEAT prints the first card in the top row of each column:
 <REPEAT Direction="Hor" Action="Duplicate" Count="5">

2: The second REPEAT does the same on each row, with a different card:
 <REPEAT Direction="Ver" Action="Increment" Count="8">

Printing pre-sorted stacks: If the attribute values are *Direction="Stack" Action="Increment"*, REPEAT puts the next Signature on the next sheet. That is, the signatures will repeat through the stack of sheets, producing a stack of pre-sorted documents.

In such applications, a Consumer may wish to print the *last* sheet first, so it ends up at the bottom of the stack. To support such applications, the optional attribute *Order="Descending"* can be used.

Nested REPEATs using Action="Increment": When multiple nested REPEATs have Action="Increment", an additional counter *d* (document counter) is applied. In every step of a REPEAT with action="Increment", *d* is incremented by 1, while *s* remains the overall sheet counter. For instance, in the following example the inner REPEAT has *Direction="Ver" Count="3"* so the Consumer will first put three signatures in a column. The outer REPEAT has *Direction="Hor" Count="4"* so the whole column will be repeated four times, incrementing the counter *d* continuously:

```
<REPEAT Action="Increment" Direction="Hor" Count="4">
  <REPEAT Action="Increment" Direction="Ver" Count="3">
    <SIGNATURE...>
  </REPEAT>
</REPEAT>
```

These are the values of *d* for the resulting sheet:

1	4	7	10
2	5	8	11
3	6	9	12

If the signature has one cell, with *PageOrder="s"*, then one should impose on the first sheet the first page of document 1, below it the first page of document 2 and so on.

When the counting of documents is incremented in the stack direction the counter *s* starts over from 1. If different documents start on the same sheet and they have different number of pages, then the next set of documents starts only after the longest document ends. For example:

```
<REPEAT Action="Increment" Direction="Hor" Count="2">
  <SIGNATURE Nrows="1" Ncols="1">
    <CELL Row="1" Col="1" PageOrder="s"/>
  </SIGNATURE>
</REPEAT>
```

If document 1 has 1 page and document 2 has 2 pages then this is the page distribution:

```
Page 1: document 1 page 1, document 2 page 1 (S=1)
Page 2:                , document 2 page 2 (S=2)
Page 3: document 3 page 1, document 4 page 1 (S=1)
```

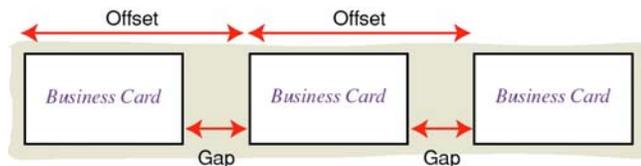
Notice also that when the count of documents imposed reaches the total count in the imposition template, the consumer keeps imposing the following documents, as if a global imaginary **REPEAT** with count "infinity" encompasses all other **REPEATS**. In other words, let N be the number of documents that a **REPEAT** and all its nested **REPEATS** consume. This is equal to the product of all *Count* attributes of the **REPEAT** (and all its nested ones), that have an attribute *Action=increment*. When all the sheets for these N documents are generated, the whole process starts again for the next N documents.

In the last example the imposition template imposes two documents (one **REPEAT** element with *count="2"*). In such a case the Consumer imposes the first two documents and then imposes the next two documents and so on until all documents are imposed.

If there is more than one **IMPOSITION** element in **SHEET_LAYOUT**, the counter *d* increments independently for each **IMPOSITION**.

Spacing of Signatures – the attributes *Spacing* and *SpacingMethod*: By default, signatures are repeated with no space between them: the TrimBox of the next Instance Document touches the TrimBox of the previous one. Optionally, the *Spacing* attribute can specify a distance between the documents.

Spacing can have two different meanings, depending on the value of another attribute, *SpacingMethod*, which has values *Gap* or *Offset*. By default, *Spacing* specifies the gap between the signatures, as shown in the illustration. If *SpacingMethod="Offset"* then *Spacing* is the distance from the *start* of one signature to the start of the next.



Multiple multi-page Instance Documents per sheet: **REPEAT** repeats a signature, which is defined as “a set of one or more pages from an Instance Document, printed on a single sheet of paper.” The business card example above shows the trivial case of a one-page Signature, where each cell equals an entire Instance Document.

It is also possible to repeat a multi-page (multi-cell) Signature on a single sheet. For instance, a personalized folded card, such as an invitation or a greeting card, could be repeated, placing two Instance Documents on each sheet using the following code:

```
<REPEAT Direction="Ver" Count="2" Action="Increment">  
  <SIGNATURE Nrows="2" Ncols="1">  
    <CELL Row="1" Col="1" PageOrder="4*s-0" Face="Up" Rotate="0"/>  
    <CELL Row="1" Col="2" PageOrder="4*s-3" Face="Up" Rotate="0"/>  
    <CELL Row="1" Col="1" PageOrder="4*s-2" Face="Dn" Rotate="0"/>  
    <CELL Row="1" Col="2" PageOrder="4*s-1" Face="Dn" Rotate="0"/>  
  </SIGNATURE>  
</REPEAT>
```

Annex A (informative) Revision history

A.1 Version 3.0 May 2010

- **Version 3.0**

Index

- <CELL> Element, iii, 19
 - <HOR_FOLD_MARKS> Element, iii, 31
 - <HOR_GUTTER> Element, iii, 28
 - <HOR_TRIM_MARKS> Element, iii, 25
 - <IMPOSITION_REF> Element, iii, 15
 - <IMPOSITION> Element, iii, 13
 - <PAGE_LAYOUT> Element, iii, 7
 - <PRINT_LAYOUT> Element, iii, 5
 - <REPEAT> Element, iii, 33
 - <SHEET_LAYOUT> Element, 10
 - <SHEET_MARK> Element, iii, 12
 - <SIGNATURE> Element, iii, 17
 - <VER_FOLD_MARKS> Element, iii, 32
 - <VER_GUTTER> Element, iii, 30
 - <VER_TRIM_MARKS> Element, iii, 27
- BleedBox, 7, 8, 9, 10, 17
- BoundingBox*, 7, 8, 9
- CIP4, 1
- HOR_FOLD_MARKS, 31
- HOR_GUTTER, 17, 28, 29, 30
- HOR_TRIM_MARKS, 25, 27
- imposing consumer, 2, 5
- Imposition, iv, 2, 3, 10, 13
- IMPOSITION, 27, 32
- IMPOSITION_REF, 3, 10, 13, 14, 15, 16
- Instance Document, 2, 10, 17, 20, 33, 35, 36
- non-imposing consumer, 5
- OCCURRENCE_REF, 12, 25, 26, 27, 31, 32
- PAGE_DESIGN, 7
- PAGE_LAYOUT, 6, 7, 8, 10, 17, 19, 24
- PageOrder*, 10, 19, 20, 21, 22, 23, 29, 35, 36
- PPML Consumer, 4, 5, 8, 9, 10, 20, 25, 34, 35
- PPML Producer, 8, 9, 12, 15, 16, 18, 20
- PRINT_LAYOUT, 2, 5, 6, 7, 10
- Production marks, 4
- REPEAT, 3, 14, 17, 33, 34, 35, 36
- Sheet layout, 3
- SHEET_LAYOUT, 2, 6, 7, 9, 10, 12, 13, 15, 20, 23, 35
- SHEET_MARK, 10, 12
- signature, 3, 8, 17, 18, 19, 20, 21, 23, 25, 26, 28, 29, 30, 31, 32, 33, 35, 36
- SIGNATURE, 3, 14, 17, 19, 20, 21, 22, 23, 25, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36
- TrimBox, 6, 7, 8, 9, 17, 19, 26, 35
- VER_FOLD_MARKS, 32
- VER_GUTTER, 17, 30
- VER_TRIM_MARKS, 25, 27