



Template Rules Overview (dsc//did/*)

Date: 31 January 2005

Creator: Michael Ferrando, Library of Congress (202-707-4454)

Stylesheet name: EAD_2002_HTML_conv-1.xsl

Language: XSLT 1.0

Objective:

Communicate to developers and EAD coders the criteria (based on grouping) which determines inline transformation of siblings in the <did>.

Proposition:

By communicating this transformation criteria, I hope to inspire the user/coder to recognize the significant benefits of being able to control the display of their data by investing in the practice and implementation of pre-determined EAD coding patterns.

Prologue:

The design of the EAD_2002_HTML_conv-1.xsl stylesheet controls the dsc//did/* elements by the XSL mode attribute. The dsc//did/* template directly follows the component template in the hierarchical transformation order of EAD elements. All dsc//did/* elements will move through this template with the exception of the dsc//did/head which is relegated to an empty template.

Template element:

```
<xsl:template match="*[not(name()='head')]" mode="dsc_did_children">
```

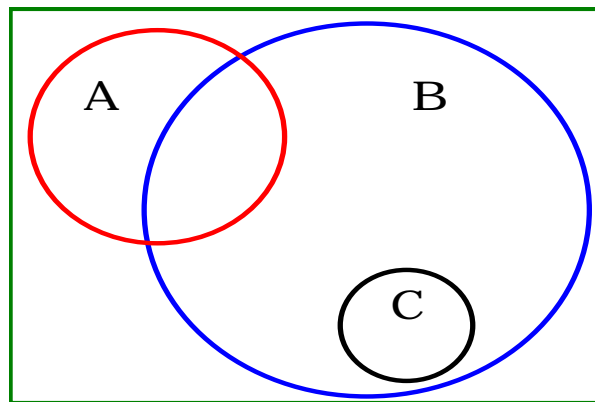
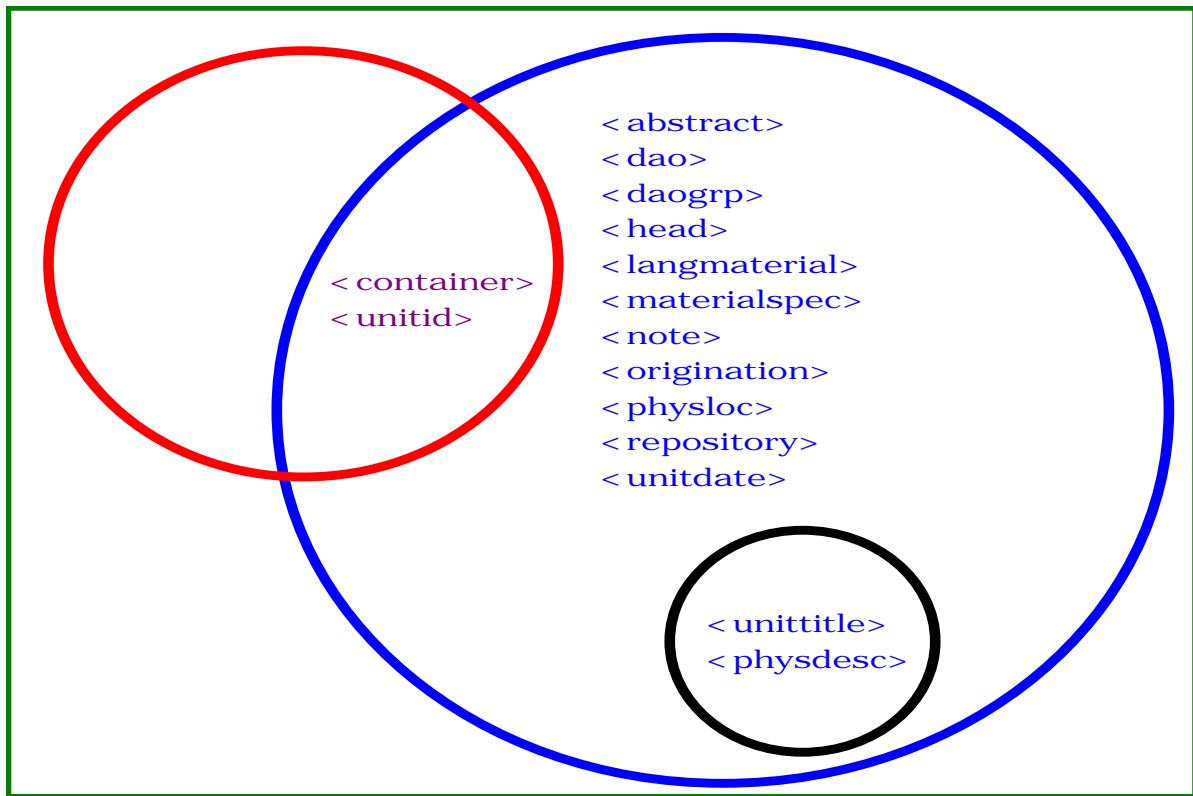
Rule:

For each did/* create a row consisting of 2 cells. Criteria will determine whether the element is transformed in cell 1 or cell 2 of the row.

Contents:

Venn Diagrams and Criteria.....	Page 2
Template Schematic.....	Page 3
Examples.....	Page 4

<DID>



Criteria:

1. If B (and Following Sibling = NULL) = Transformation in Cell 2
2. If B (and not A and not C) = Transform in Cell 2
3. If C = Transform in Cell 2
4. If A (and Following Sibling = B and not C) = Transformation in Cell 2
5. If A (and Following Sibling = C) = Transformation in Cell 1
6. If A (and Following Sibling = A) = Hold current() and check Following Sibling at condition 3

Template Schematic:

```
1   Open template
2   Variable: create DOM string of elements = $y
3   Variable: position of current() in parent::did = $p
4   Variable: get substring-before current() = $x
5   [1] Open choose
6       Open when test current() = Cell 1
7           NULL
8       Close when
9       Open otherwise
10          Open ROW
11             Open Cell 1
12                Open IF test current() = Group C
13                   Check preceding sibling against criteria
14                   Transform preceding sibling(s) = true = Cell 1
15                Close IF
16             Close Cell 1
17             Open Cell 2
18                [2] Open choose
19                   Open when test current() = Group A or Group C
20                      [apply local formatting preferences]
21                   Close when
22                   Open otherwise
23                      Transform parent::did/*[$p]
24                   Close otherwise
25                Close choose
26             Close Cell 2
27          Close Row
28       Close otherwise
29   Close choose
30   Close template
```

Comments:

Line 2: DOM string for each did/* return = concat(name(), '*', st(name), '%', ct(did::child), ';') = unitid*1006%1007;

Line 5-29: choose [1] restricts the template to generating markup in conformance to the pattern

Line 6-8: when check excludes cell 1 elements which will be transformed by Group C siblings

Line 9-28: otherwise generates row

Line 11-16: cell 1 generates all Group A preceding siblings of any current() Group C element

Line 18-25: choose [2] determines particular formatting preferences of cell 2 elements

Line 22-24: otherwise transforms generic elements

Examples:

Example 1 Source

```
<did>
  <unittitle encodinganalog="245$a">Notebooks </unittitle>
  <unitid>(96-97)</unitid>
  <physdesc>
    <extent encodinganalog="300">(2 folders)</extent>
  </physdesc>
</did>
```

Example 1 HTML

```
<TR>
  <TD VALIGN="top" COLSPAN="3"></TD>
  <TD VALIGN="top" COLSPAN="11">Notebooks </TD>
</TR>
<TR>
  <TD VALIGN="top" COLSPAN="4">(96-97)</TD>
  <TD VALIGN="top" COLSPAN="10">(2 folders)</TD>
</TR>
```

Example 1 Output

	Notebooks
(96-97)	(2 folders)

Example 2 Source

```
<did>
  <unittitle encodinganalog="245$a">Set III, 1967</unittitle>
  <physdesc>
    <extent encodinganalog="300">(3 folders)</extent>
  </physdesc>
  <container type="box">I:58</container>
  <physdesc>
    <extent encodinganalog="300">(2 folders)</extent>
  </physdesc>
</did>
```

Example 2 HTML

```
<TR>
  <TD VALIGN="top" COLSPAN="2"></TD>
  <TD VALIGN="top" COLSPAN="12">Set III, 1967</TD>
</TR>
<TR>
  <TD VALIGN="top" COLSPAN="3"></TD>
  <TD VALIGN="top" COLSPAN="11">(3 folders)</TD>
</TR>
<TR>
  <TD VALIGN="top" COLSPAN="3"><B><SMALL>BOX </SMALL></B>I:58</TD>
  <TD VALIGN="top" COLSPAN="11">(2 folders)</TD>
</TR>
```

Example 2 Output

	Set III, 1967
	(3 folders)
BOX I:58	(2 folders)

Example 3 Source

```
<did>
  <container type="box">271</container>
  <container type="folder">1-5</container>
  <unittitle encodinganalog="245$a">National Broiler Mktg. Ass'n v.
    United States </unittitle>
  <physdesc>
    <extent encodinganalog="300">(1 folder)</extent>
  </physdesc>
  <container type="box">272</container>
  <container type="folder">1-10</container>
  <physdesc>
    <extent encodinganalog="300">(1 folder)</extent>
  </physdesc>
  <unitid encodinganalog="050" label="Call no.:">LOT 13074</unitid>
</did>
```

Example 3 HTML

```
<TR>
  <TD VALIGN="top" COLSPAN="2">
    <B><SMALL> BOX </SMALL></B>271
    <BR><B><SMALL> FOLDER </SMALL></B>1-5
  </TD>
  <TD VALIGN="top" COLSPAN="12">National Broiler Mktg. Ass'n v.
    United States </TD>
</TR>
<TR>
  <TD VALIGN="top" COLSPAN="3"></TD>
  <TD VALIGN="top" COLSPAN="11">(1 folder)</TD>
</TR>
<TR>
  <TD VALIGN="top" COLSPAN="3">
    <B><SMALL>BOX </SMALL></B>272
    <BR><B><SMALL> FOLDER </SMALL></B>1-10
  </TD>
  <TD VALIGN="top" COLSPAN="11">(1 folder)</TD>
</TR>
<TR>
  <TD VALIGN="top" COLSPAN="3"></TD>
  <TD VALIGN="top" COLSPAN="11">
    <B><SMALL>CALL NO.: </SMALL></B>LOT 13074
  </TD>
</TR>
```

Example 3 Output

BOX 271	National Broiler Mktg. Ass'n v. United States
FOLDER 1-5	(1 folder)
BOX 272	(1 folder)
FOLDER 1-10	CALL NO.: LOT 13074