Morpho - Bug #4426

export fails to create html with attribute information

09/30/2009 09:06 AM - Matt Jones

Status:	Resolved	**Start date:**	09/30/2009
Priority:	Normal	**Due date:**	
Assignee:	Jing Tao	**% Done:**	0%
Category:	morpho - general	**Estimated time:**	0.00 hour
Target version:	1.8.0	**Spent time:**	0.00 hour
Bugzilla-Id:	4426		

Description

Hi Gail,

Yes, my guess is that this is a stylesheet problem. The stylesheet has two modes -- one that displays just the summary information and attribute information separately, and one that displays them together. It appears for some reason that the export function is not using the mode that displays all of the metadata together. We'll look into this and get back to you.

Matt

On Wed Sep 30 06:17:29 2009, gss1@cornell.edu wrote:

> Greetings,
>
> I notice that when I export a new data package from Morpho
> (v1.7.0-RC2), the HTML display of the metadata record doesn't include
> complete metadata for individual tables (or links to complete
> metadata), even though the information is in the XML file. There is
> some partial/summary table info at the top of the HTML display of the
> record, but no attribute info. Is that a stylesheet problem, and is a
> fix in the works? We have researchers who are sharing data sets with
> colleagues for whom that HTML display is very useful. Any thoughts on
> why the display has become a little whacky?
>
> I recreated the problem by creating a new/clean data set with just
> one table - I can send a screenshot and/or the metadata file, if
> that's helpful.
>
> Thanks so much,
> Gail
>
> Gail Steinhart
> Research Data & Environmental Sciences Librarian
> Albert R. Mann Library
> Cornell University
> Ithaca, NY 14853
>
> Phone: 607-255-7251
> Fax: 607-255-0318
> E-mail: GSS1@cornell.edu

History

#1 - 09/30/2009 11:25 AM - Jing Tao

The condition seems backwards to me.ï¿½ The reason it is in there is so thatthe same stylesheet can be used to create the HTML for display in the UI and in the export.ï¿½ In the UI, the top-level metadata is in a separate pane than the entity and attribute metadata, and they each have their own mode.ï¿½ In the export, the mode seems like it should be printall, and so the condition should be changed to $displaymodule=='printall'.ï¿½ But I'm not sure -- I think you'll need to carefully go through the logic of the conditionals to make sure all of the cases are handled properly, then make the needed changes.

Matt

On Wed, Sep 30, 2009 at 9:44 AM, Jing Tao <tao@nceas.ucsb.edu> wrote:
Hi, Matt:

```
Your guess is right. But I think this is a bug in eml stylesheet.

In eml-attrubte.xsl code, there is a condition line:
      &lt;xsl:when test="$displaymodule!='printall'"&gt;
      ï¿½.....
      ï¿½&lt;xsl:call-template name="attributecommonvertical"&gt;
      ï¿½.....
      &lt;/xsl:when&gt;

This means when "displaymodule" does NOT equals "printall", the style sheet will display the attributes. I cou
ldn't
      understand why we do this.

During the export action of morpho, the value of "displaymodule" is overwritten by "printall". So the html pag
e doesn't
      have the attribute info.

I guess we should remove the condition line. Do you see any problem?

Thanks,

Jing
```

#2 - 12/08/2009 03:35 PM - Jing Tao

```
<table xsl:use-attribute-sets="cellspacing" class="{$tableattributeStyle}" width="100%">
<xsl:choose>
<xsl:when test="$displaymodule!='printall'">
<xsl:choose>
<xsl:when test="references!="">
<xsl:variable name="ref_id" select="references"/>
<xsl:variable name="references" select="$ids[@id=$ref_id]" />
<xsl:for-each select="$references">
<!-- changed template name from attributecommon to attributecommonvertical -->
<xsl:call-template name="attributecommonvertical">
<xsl:with-param name="docid" select="$docid"/>
<xsl:with-param name="entitytype" select="$entitytype"/>
<xsl:with-param name="entityindex" select="$entityindex"/>
</xsl:call-template>
</xsl:for-each>
</xsl:when>
<xsl:otherwise>
<xsl:call-template name="attributecommonvertical">
<xsl:with-param name="docid" select="$docid"/>
<xsl:with-param name="entitytype" select="$entitytype"/>
<xsl:with-param name="entityindex" select="$entityindex"/>
</xsl:call-template>
</xsl:otherwise>
</xsl:choose>
</xsl:when>
<!-- Comment out redundent call
<xsl:otherwise>
<xsl:choose>
<xsl:when test="references!="">
<xsl:variable name="ref_id" select="references"/>
<xsl:variable name="references" select="$ids[@id=$ref_id]" />
<xsl:for-each select="$references">
<xsl:call-template name="attributecommonvertical">
<xsl:with-param name="docid" select="$docid"/>
<xsl:with-param name="entitytype" select="$entitytype"/>
<xsl:with-param name="entityindex" select="$entityindex"/>
</xsl:call-template>
</xsl:for-each>
</xsl:when>
<xsl:otherwise>
<xsl:call-template name="attributecommonvertical">
<xsl:with-param name="docid" select="$docid"/>
<xsl:with-param name="entitytype" select="$entitytype"/>
<xsl:with-param name="entityindex" select="$entityindex"/>
</xsl:call-template>
</xsl:otherwise>
</xsl:choose>
```

```
</xsl:otherwise>
-->
</xsl:choose>
</table>
```

Above is our current code in attribute.xsl.

Someone (maybe myself) comment out the redundant code in the otherwise part. However, it left the condition - displaymodule!=printall. So when displaymodule=printall, it will do nothing.

Since in original code, the displaymodule equals or not equal printall will do same action, so we can remove the condition and redundant code.
So code will be:

```
<table xsl:use-attribute-sets="cellspacing" class="{$tableattributeStyle}" width="100%">
<xsl:choose>
<xsl:when test="references!="">
<xsl:variable name="ref_id" select="references"/>
<xsl:variable name="references" select="$ids[@id=$ref_id]" />
<xsl:for-each select="$references">
<!-- changed template name from attributecommon to attributecommonvertical -->
<xsl:call-template name="attributecommonvertical">
<xsl:with-param name="docid" select="$docid"/>
<xsl:with-param name="entitytype" select="$entitytype"/>
<xsl:with-param name="entityindex" select="$entityindex"/>
</xsl:call-template>
</xsl:for-each>
</xsl:when>
<xsl:otherwise>
<xsl:call-template name="attributecommonvertical">
<xsl:with-param name="docid" select="$docid"/>
<xsl:with-param name="entitytype" select="$entitytype"/>
<xsl:with-param name="entityindex" select="$entityindex"/>
</xsl:call-template>
</xsl:otherwise>
</xsl:choose>
</table>
```

After the changing, the import and morpho display work good.

#3 - 12/10/2009 08:41 AM - Jing Tao

This bug was fixed base on above solution.

#4 - 03/27/2013 02:26 PM - Redmine Admin

Original Bugzilla ID was 4426