

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <!-- $Id: MARC21slim2DC.xsl,v 1.1 2003/01/06 08:20:27 adam Exp $ -->
3  <!DOCTYPE stylesheet>
4  <xsl:stylesheet version="1.0"
5     xmlns:marc="http://www.loc.gov/MARC21/slim"
6     xmlns:items="http://www.koha-community.org/items"
7     xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
8     xmlns:exsl="http://exslt.org/common"
9     xmlns:str="http://exslt.org/strings"
10    exclude-result-prefixes="marc items str" extension-element-prefixes="exsl">
11    <xsl:import href="MARC21slimUtils.xsl"/>
12    <xsl:output method="html" indent="yes" omit-xml-declaration="yes" encoding="
13      UTF-8"/>
14
15    <xsl:key name="item-by-status" match="items:item" use="items:status"/>
16
17    <xsl:template match="/">
18      <xsl:apply-templates/>
19    </xsl:template>
20
21    <xsl:template match="marc:record">
22
23      <xsl:variable name="itemcount" select="count(items:items/items:item)"/>
24
25      <!-- Option: Display Alternate Graphic Representation (MARC 880) -->
26      <xsl:variable name="display880" select="boolean(marc:datafield[@tag=880])"/>
27
28      <xsl:variable name="OPACResultsMaxItems" select=
29        "number(marc:sysprefs/marc:syspref[@name='OPACResultsMaxItems']+0)"/>
30      <xsl:variable name="OPACResultsMaxItemsUnavailable" select=
31        "number(marc:sysprefs/marc:syspref[@name='OPACResultsMaxItemsUnavailable']+0)"/>
32      <xsl:variable name="OPACResultsUnavailableGroupingBy" select=
33        "marc:sysprefs/marc:syspref[@name='OPACResultsUnavailableGroupingBy']"/>
34      <xsl:variable name="UseControlNumber" select=
35        "marc:sysprefs/marc:syspref[@name='UseControlNumber']"/>
36      <xsl:variable name="UseAuthoritiesForTracings" select=
37        "marc:sysprefs/marc:syspref[@name='UseAuthoritiesForTracings']"/>
38      <xsl:variable name="OPACResultsLibrary" select=
39        "marc:sysprefs/marc:syspref[@name='OPACResultsLibrary']"/>
40      <xsl:variable name="hidelostitems" select=
41        "marc:sysprefs/marc:syspref[@name='hidelostitems']"/>
42      <xsl:variable name="DisplayOPACiconsXSLT" select=
43        "marc:sysprefs/marc:syspref[@name='DisplayOPACiconsXSLT']"/>
44      <xsl:variable name="OPACURLOpenInNewWindow" select=
45        "marc:sysprefs/marc:syspref[@name='OPACURLOpenInNewWindow']"/>
46      <xsl:variable name="URLLinkText" select=
47        "marc:sysprefs/marc:syspref[@name='URLLinkText']"/>
48      <xsl:variable name="Show856uAsImage" select=
49        "marc:sysprefs/marc:syspref[@name='OPACDisplay856uAsImage']"/>
50      <xsl:variable name="AlternateHoldingsField" select=
51        "substring(marc:sysprefs/marc:syspref[@name='AlternateHoldingsField'], 1, 3)"/>
52      <xsl:variable name="AlternateHoldingsSubfields" select=
53        "substring(marc:sysprefs/marc:syspref[@name='AlternateHoldingsField'], 4)"/>
54      <xsl:variable name="AlternateHoldingsSeparator" select=
55        "marc:sysprefs/marc:syspref[@name='AlternateHoldingsSeparator']"/>
56      <xsl:variable name="OPACItemLocation" select=
57        "marc:sysprefs/marc:syspref[@name='OPACItemLocation']"/>
58      <xsl:variable name="OPACTrackClicks" select=
59        "marc:sysprefs/marc:syspref[@name='TrackClicks']"/>
60      <xsl:variable name="BiblioDefaultView" select=
61        "marc:sysprefs/marc:syspref[@name='BiblioDefaultView']"/>
62      <xsl:variable name="leader" select="marc:leader"/>
63      <xsl:variable name="leader6" select="substring($leader,7,1)"/>
64      <xsl:variable name="leader7" select="substring($leader,8,1)"/>
65      <xsl:variable name="leader19" select="substring($leader,20,1)"/>
66      <xsl:variable name="biblionumber" select=
67        "marc:datafield[@tag=999]/marc:subfield[@code='c']"/>
68      <xsl:variable name="isbn" select=

```

```

49 "marc:datafield[@tag=020]/marc:subfield[@code='a']" />
50 <xsl:variable name="controlField008" select="marc:controlfield[@tag=008]" />
51 <xsl:variable name="typeOf008">
52   <xsl:choose>
53     <xsl:when test="$leader19='a'">ST</xsl:when>
54     <xsl:when test="$leader6='a'">
55       <xsl:choose>
56         <xsl:when test="$leader7='a' or $leader7='c' or $leader7='d' or
57           $leader7='m'">BK</xsl:when>
58         <xsl:when test="$leader7='b' or $leader7='i' or $leader7='s'">CR
59       </xsl:when>
60     </xsl:choose>
61   </xsl:when>
62   <xsl:when test="$leader6='t'">BK</xsl:when>
63   <xsl:when test="$leader6='o' or $leader6='p'">MX</xsl:when>
64   <xsl:when test="$leader6='m'">CF</xsl:when>
65   <xsl:when test="$leader6='e' or $leader6='f'">MP</xsl:when>
66   <xsl:when test="$leader6='g' or $leader6='k' or $leader6='r'">VM
67   </xsl:when>
68   <xsl:when test="$leader6='i' or $leader6='j'">MU</xsl:when>
69   <xsl:when test="$leader6='c' or $leader6='d'">PR</xsl:when>
70 </xsl:choose>
71 </xsl:variable>
72 <xsl:variable name="controlField008-23" select="substring($controlField008,24,1)"
73 />
74 <xsl:variable name="controlField008-21" select="substring($controlField008,22,1)"
75 />
76 <xsl:variable name="controlField008-22" select="substring($controlField008,23,1)"
77 />
78 <xsl:variable name="controlField008-24" select="substring($controlField008,25,4)"
79 />
80 <xsl:variable name="controlField008-26" select="substring($controlField008,27,1)"
81 />
82 <xsl:variable name="controlField008-29" select="substring($controlField008,30,1)"
83 />
84 <xsl:variable name="controlField008-34" select="substring($controlField008,35,1)"
85 />
86 <xsl:variable name="controlField008-33" select="substring($controlField008,34,1)"
87 />
88 <xsl:variable name="controlField008-30-31" select="
89   substring($controlField008,31,2)" />
90
91 <xsl:variable name="physicalDescription">
92   <xsl:if test="$typeOf008='CF' and
93     marc:controlfield[@tag=007][substring(.,12,1)='a']">
94     reformatted digital
95   </xsl:if>
96   <xsl:if test="$typeOf008='CF' and
97     marc:controlfield[@tag=007][substring(.,12,1)='b']">
98     digitized microfilm
99   </xsl:if>
100   <xsl:if test="$typeOf008='CF' and
101     marc:controlfield[@tag=007][substring(.,12,1)='d']">
102     digitized other analog
103   </xsl:if>
104
105   <xsl:variable name="check008-23">
106     <xsl:if test="$typeOf008='BK' or $typeOf008='MU' or $typeOf008='CR' or
107       $typeOf008='MX'">
108       <xsl:value-of select="true()"></xsl:value-of>
109     </xsl:if>
110   </xsl:variable>
111   <xsl:variable name="check008-29">
112     <xsl:if test="$typeOf008='MP' or $typeOf008='VM'">
113       <xsl:value-of select="true()"></xsl:value-of>
114     </xsl:if>
115   </xsl:variable>

```

```

99     <xsl:choose>
100       <xsl:when test="($check008-23 and $controlField008-23='f') or
101         ($check008-29 and $controlField008-29='f')">
102         braille
103       </xsl:when>
104       <xsl:when test="($controlField008-23=' ' and ($leader6='c' or
105         $leader6='d')) or (($typeOf008='BK' or $typeOf008='CR') and
106         ($controlField008-23=' ' or $controlField008='r'))">
107         print
108       </xsl:when>
109       <xsl:when test="$leader6 = 'm' or ($check008-23 and
110         $controlField008-23='s') or ($check008-29 and $controlField008-29='s')">
111         electronic
112       </xsl:when>
113       <xsl:when test="($check008-23 and $controlField008-23='b') or
114         ($check008-29 and $controlField008-29='b')">
115         microfiche
116       </xsl:when>
117       <xsl:when test="($check008-23 and $controlField008-23='a') or
118         ($check008-29 and $controlField008-29='a')">
119         microfilm
120       </xsl:when>
121       <xsl:when test="($controlField008-23='d' and ($typeOf008='BK' or
122         $typeOf008='CR'))">
123         large print
124       </xsl:when>
125     </xsl:choose>
126
127 <!--
128
129 <xsl:if test="marc:datafield[@tag=130]/marc:subfield[@code='h']">
130   <xsl:call-template name="chopBrackets">
131     <xsl:with-param name="chopString">
132       <xsl:value-of
133         select="marc:datafield[@tag=130]/marc:subfield[@code='h']"></
134       xsl:value-of>
135     </xsl:with-param>
136   </xsl:call-template>
137 </xsl:if>
138
139 <xsl:if test="marc:datafield[@tag=240]/marc:subfield[@code='h']">
140   <xsl:call-template name="chopBrackets">
141     <xsl:with-param name="chopString">
142       <xsl:value-of
143         select="marc:datafield[@tag=240]/marc:subfield[@code='h']"></
144       xsl:value-of>
145     </xsl:with-param>
146   </xsl:call-template>
147 </xsl:if>
148
149 <xsl:if test="marc:datafield[@tag=242]/marc:subfield[@code='h']">
150   <xsl:call-template name="chopBrackets">
151     <xsl:with-param name="chopString">
152       <xsl:value-of
153         select="marc:datafield[@tag=242]/marc:subfield[@code='h']"></
154       xsl:value-of>
155     </xsl:with-param>
156   </xsl:call-template>
157 </xsl:if>
158
159 <xsl:if test="marc:datafield[@tag=245]/marc:subfield[@code='h']">
160   <xsl:call-template name="chopBrackets">
161     <xsl:with-param name="chopString">
162       <xsl:value-of
163         select="marc:datafield[@tag=245]/marc:subfield[@code='h']"></
164       xsl:value-of>
165     </xsl:with-param>
166   </xsl:call-template>
167 </xsl:if>
168
169 <xsl:if test="marc:datafield[@tag=246]/marc:subfield[@code='h']">
170   <xsl:call-template name="chopBrackets">
171     <xsl:with-param name="chopString">
172       <xsl:value-of
173         select="marc:datafield[@tag=246]/marc:subfield[@code='h']"></
174       xsl:value-of>
175     </xsl:with-param>
176   </xsl:call-template>
177 </xsl:if>

```

```

151         <xsl:value-of
           select="marc:datafield[@tag=246]/marc:subfield[@code='h']"></
           xsl:value-of>
152     </xsl:with-param>
153 </xsl:call-template>
154 </xsl:if>
155 <xsl:if test="marc:datafield[@tag=730]/marc:subfield[@code='h']">
156     <xsl:call-template name="chopBrackets">
157         <xsl:with-param name="chopString">
158             <xsl:value-of
               select="marc:datafield[@tag=730]/marc:subfield[@code='h']"></
               xsl:value-of>
159         </xsl:with-param>
160     </xsl:call-template>
161 </xsl:if>
162 <xsl:for-each select="marc:datafield[@tag=256]/marc:subfield[@code='a']">
163     <xsl:value-of select="."></xsl:value-of>
164 </xsl:for-each>
165 <xsl:for-each
   select="marc:controlfield[@tag=007][substring(text(),1,1)='c']">
166     <xsl:choose>
167         <xsl:when test="substring(text(),14,1)='a'">
168             access
169         </xsl:when>
170         <xsl:when test="substring(text(),14,1)='p'">
171             preservation
172         </xsl:when>
173         <xsl:when test="substring(text(),14,1)='r'">
174             replacement
175         </xsl:when>
176     </xsl:choose>
177 </xsl:for-each>
178 -->
179 <!-- Modified for authcat: Material type section removed -->
180 <!--
181     <xsl:for-each
   select="marc:datafield[@tag=856]/marc:subfield[@code='q'][string-length(.)>1]
   ">
182         <xsl:value-of select="."></xsl:value-of>
183     </xsl:for-each>
184     <xsl:for-each select="marc:datafield[@tag=300]">
185         <xsl:call-template name="subfieldSelect">
186             <xsl:with-param name="codes">abce</xsl:with-param>
187         </xsl:call-template>
188     </xsl:for-each>
189 -->
190 </xsl:variable>
191
192 <!-- Title Statement: Alternate Graphic Representation (MARC 880) -->
193 <xsl:if test="$display880">
194     <xsl:call-template name="m880Select">
195         <xsl:with-param name="basetags">245</xsl:with-param>
196         <xsl:with-param name="codes">abhfgknps</xsl:with-param>
197         <xsl:with-param name="bibno"><xsl:value-of select="$biblionumber"/>
198     </xsl:call-template>
199 </xsl:if>
200
201 <a>
202     <xsl:attribute name="href">
203         <xsl:call-template name="buildBiblioDefaultViewURL">
204             <xsl:with-param name="BiblioDefaultView">
205                 <xsl:value-of select="$BiblioDefaultView"/>
206             </xsl:with-param>
207         </xsl:call-template>
208         <xsl:value-of select="str:encode-uri($biblionumber, true())"/>
209     </xsl:attribute>

```

```

210     <xsl:attribute name="class"><xsl:value-of select="'title'" /></xsl:attribute>
211
212     <xsl:if test="marc:datafield[@tag=245]">
213         <xsl:for-each select="marc:datafield[@tag=245]">
214             <xsl:call-template name="subfieldSelect">
215                 <xsl:with-param name="codes">a</xsl:with-param>
216             </xsl:call-template>
217             <xsl:text> </xsl:text>
218             <!-- 13381 add additional subfields-->
219             <!-- bz 17625 adding subfields f and g -->
220             <xsl:for-each select="marc:subfield[contains('bcfghknps', @code)]">
221                 <xsl:choose>
222                     <xsl:when test="@code='h'">
223                         <!-- 13381 Span class around subfield h so it can be
224                             suppressed via css -->
225                         <span class="title_medium"><xsl:apply-templates/> <xsl:text>
226                             </xsl:text> </span>
227                     </xsl:when>
228                     <xsl:when test="@code='c'">
229                         <!-- 13381 Span class around subfield c so it can be
230                             suppressed via css -->
231                         <span class="title_resp_stmt"><xsl:apply-templates/>
232                             <xsl:text> </xsl:text> </span>
233                     </xsl:when>
234                     <xsl:otherwise>
235                         <xsl:apply-templates/>
236                         <xsl:text> </xsl:text>
237                     </xsl:otherwise>
238                 </xsl:choose>
239             </xsl:for-each>
240         </xsl:if>
241     </a>
242
243     <!-- OpenURL -->
244     <xsl:variable name="OPACShowOpenURL" select=
245         "marc:sySPrefs/marc:sySPref[@name='OPACShowOpenURL']" />
246     <xsl:variable name="OpenURLImageLocation" select=
247         "marc:sySPrefs/marc:sySPref[@name='OpenURLImageLocation']" />
248     <xsl:variable name="OpenURLText" select=
249         "marc:sySPrefs/marc:sySPref[@name='OpenURLText']" />
250     <xsl:variable name="OpenURLResolverURL" select=
251         "marc:variables/marc:variable[@name='OpenURLResolverURL']" />
252
253     <xsl:if test="$OPACShowOpenURL = 1 and $OpenURLResolverURL != ''">
254         <xsl:variable name="openurltext">
255             <xsl:choose>
256                 <xsl:when test="$OpenURLText != ''">
257                     <xsl:value-of select="$OpenURLText" />
258                 </xsl:when>
259                 <xsl:otherwise>
260                     <xsl:text>OpenURL</xsl:text>
261                 </xsl:otherwise>
262             </xsl:choose>
263         </xsl:variable>
264
265         <div class="results_summary"><a>
266             <xsl:attribute name="href">
267                 <xsl:value-of select="$OpenURLResolverURL" />
268             </xsl:attribute>
269             <xsl:attribute name="title">
270                 <xsl:value-of select="$openurltext" />
271             </xsl:attribute>
272             <xsl:attribute name="class">
273                 <xsl:text>OpenURL</xsl:text>
274             </xsl:attribute>
275             <xsl:if test="$OPACURLOpenInNewWindow='1'">

```

```

269     <xsl:attribute name="target">
270       <xsl:text>_blank</xsl:text>
271     </xsl:attribute>
272   </xsl:if>
273   <xsl:choose>
274     <xsl:when test="$OpenURLImageLocation != ''">
275       <img>
276         <xsl:attribute name="src">
277           <xsl:value-of select="$OpenURLImageLocation" />
278         </xsl:attribute>
279       </img>
280     </xsl:when>
281     <xsl:otherwise>
282       <xsl:value-of select="$openurltext" />
283     </xsl:otherwise>
284   </xsl:choose>
285 </a></div>
286 </xsl:if>
287 <!-- End of OpenURL -->
288
289 <!-- Author Statement: Alternate Graphic Representation (MARC 880) -->
290 <xsl:if test="$display880">
291   <xsl:call-template name="m880Select">
292     <xsl:with-param name="basetags">100,110,111,700,710,711</xsl:with-param>
293     <xsl:with-param name="codes">abc</xsl:with-param>
294   </xsl:call-template>
295 </xsl:if>
296
297 <xsl:choose>
298 <xsl:when test="marc:datafield[@tag=100] or marc:datafield[@tag=110] or
marc:datafield[@tag=111] or marc:datafield[@tag=700] or marc:datafield[@tag=710] or
marc:datafield[@tag=711]">
299 <!-- Modified for authcat: "MedStar authors:-->
300   <span class="byAuthor">MedStar authors: </span>
301   <ul class="author_resource_list">
302     <!-- #13383 -->
303     <xsl:for-each select="marc:datafield[(@tag=100 or @tag=700 or @tag=110 or
@tag=710 or @tag=111 or @tag=711) and @ind1!='z']">
304       <li>
305         <xsl:call-template name="chopPunctuation">
306           <xsl:with-param name="chopString">
307             <xsl:call-template name="subfieldSelect">
308               <xsl:with-param name="codes">
309                 <xsl:choose>
310                   <!-- #13383 include subfield e for field 111 -->
311                   <xsl:when test="@tag=111 or @tag=711">a</xsl:when>
312                   <xsl:when test="@tag=110 or @tag=710">a</xsl:when>
313                   <xsl:otherwise>a</xsl:otherwise>
314                 </xsl:choose>
315               </xsl:with-param>
316             </xsl:call-template>
317           </xsl:with-param>
318           <xsl:with-param name="punctuation">
319             <xsl:text>:,;/ </xsl:text>
320           </xsl:with-param>
321         </xsl:call-template>
322         <!-- Display title portion for 110 and 710 fields -->
323         <!--
324         <xsl:if test="(@tag=110 or @tag=710) and boolean(marc:subfield[@code='c' or
@code='d' or @code='n' or @code='t'])">
325           <span class="titleportion">
326             <xsl:choose>
327               <xsl:when test="marc:subfield[@code='c' or @code='d' or
@code='n'] [not (marc:subfield[@code='t'])]"><xsl:text>
328               </xsl:text></xsl:when>
329               <xsl:otherwise><xsl:text>. </xsl:text></xsl:otherwise>

```

```

330         <xsl:call-template name="chopPunctuation">
331             <xsl:with-param name="chopString">
332                 <xsl:call-template name="subfieldSelect">
333                     <xsl:with-param name="codes">cdnt</xsl:with-param>
334                 </xsl:call-template>
335             </xsl:with-param>
336         </xsl:call-template>
337     </span>
338 </xsl:if>
339 -->
340 <!-- Display title portion for 111 and 711 fields -->
341 <xsl:if test="(@tag=111 or @tag=711) and boolean(marc:subfield[@code='c' or
342 @code='d' or @code='g' or @code='n' or @code='t'])">
343     <span class="titleportion">
344         <xsl:choose>
345             <xsl:when test="marc:subfield[@code='c' or @code='d' or
346 @code='g' or @code='n'][not(marc:subfield[@code='t'])]"
347             ><xsl:text> </xsl:text></xsl:when>
348             <xsl:otherwise><xsl:text>. </xsl:text></xsl:otherwise>
349         </xsl:choose>
350
351         <xsl:call-template name="chopPunctuation">
352             <xsl:with-param name="chopString">
353                 <xsl:call-template name="subfieldSelect">
354                     <xsl:with-param name="codes">cdgnt</xsl:with-param>
355                 </xsl:call-template>
356             </xsl:with-param>
357         </xsl:call-template>
358     </span>
359 </xsl:if>
360 <!-- Display dates for 100 and 700 fields -->
361 <!--
362 <xsl:if test="(@tag=100 or @tag=700) and marc:subfield[@code='d']">
363     <span class="authordates">
364         <xsl:text>,</xsl:text>
365         <xsl:call-template name="chopPunctuation">
366             <xsl:with-param name="chopString">
367                 <xsl:call-template name="subfieldSelect">
368                     <xsl:with-param name="codes">d</xsl:with-param>
369                 </xsl:call-template>
370             </xsl:with-param>
371         </xsl:call-template>
372     </span>
373 </xsl:if>
374 -->
375 <!-- Display title portion for 100 and 700 fields -->
376 <xsl:if test="@tag=700 and marc:subfield[@code='t']">
377     <span class="titleportion">
378         <xsl:text>.</xsl:text>
379         <xsl:call-template name="chopPunctuation">
380             <xsl:with-param name="chopString">
381                 <xsl:call-template name="subfieldSelect">
382                     <xsl:with-param name="codes">t</xsl:with-param>
383                 </xsl:call-template>
384             </xsl:with-param>
385         </xsl:call-template>
386     </span>
387 </xsl:if>
388 <!-- Display relators for 1XX and 7XX fields (code e replaced with 4 -->
389 <xsl:if test="marc:subfield[@code='4' or
390 @code='4'] [not(parent::*[@tag=111])] or (self::*[@tag=111] and
391 marc:subfield[@code='4' or @code='j'] [. != ''])">
392     <span class="relatorcode">
393         <xsl:text> [</xsl:text>
394         <xsl:choose>
395             <xsl:when test="@tag=111 or @tag=711">
396                 <xsl:choose>

```



```

392         <!-- Prefer j over 4 for 111 and 711 -->
393
394         <xsl:when test="marc:subfield[@code='j']">
395             <xsl:for-each select="marc:subfield[@code='j']">
396                 <xsl:value-of select="."/>
397                 <xsl:if test="position() != last()"/>, </xsl:if>
398             </xsl:for-each>
399         </xsl:when>
400         <xsl:otherwise>
401             <xsl:for-each select="marc:subfield[@code=4]">
402                 <xsl:value-of select="."/>
403                 <xsl:if test="position() != last()"/>, </xsl:if>
404             </xsl:for-each>
405         </xsl:otherwise>
406     </xsl:choose>
407 </xsl:when>
408 <!-- Prefer e over 4 on 100 and 110 -->
409
410     <xsl:when test="marc:subfield[@code='e']">
411         <xsl:for-each select="marc:subfield[@code='4'] [not(@tag=111)
412             or not(@tag=711)]">
413             <xsl:value-of select="."/>
414             <xsl:if test="position() != last()"/>, </xsl:if>
415         </xsl:for-each>
416     </xsl:when>
417     <xsl:otherwise>
418         <xsl:for-each select="marc:subfield[@code=4]">
419             <xsl:value-of select="."/>
420             <xsl:if test="position() != last()"/>, </xsl:if>
421         </xsl:for-each>
422     </xsl:otherwise>
423 </xsl:choose>
424     <xsl:text>]</xsl:text>
425 </span>
426 </xsl:if>
427
428 </li>
429 </xsl:for-each>
430 </ul>
431 </xsl:when>
432 </xsl:choose>
433
434 <xsl:call-template name="show-series">
435     <xsl:with-param name="searchurl">/cgi-bin/koha/opac-search.pl</xsl:with-param>
436     <xsl:with-param name="UseControlNumber" select="$UseControlNumber"/>
437     <xsl:with-param name="UseAuthoritiesForTracings" select=
438         "$UseAuthoritiesForTracings"/>
439 </xsl:call-template>
440
441 <xsl:if test="marc:datafield[@tag=250]">
442 <div class="results_summary edition">
443 <span class="label">Edition: </span>
444     <xsl:for-each select="marc:datafield[@tag=250]">
445         <xsl:call-template name="subfieldSelect">
446             <xsl:with-param name="codes">ab</xsl:with-param>
447         </xsl:call-template>
448     </xsl:for-each>
449 </div>
450 </xsl:if>
451
452 <xsl:if test="marc:datafield[@tag=773]">
453     <xsl:for-each select="marc:datafield[@tag=773]">
454         <xsl:if test="marc:subfield[@code='t']">
455 <div class="results_summary source">
456 <span class="label">Source: </span>
457     <xsl:value-of select="marc:subfield[@code='t']"/>

```



```

457     </div>
458         </xsl:if>
459     </xsl:for-each>
460 </xsl:if>
461
462 <xsl:if test="$DisplayOPACIconsXSLT!='0'">
463     <div class="results_summary type">
464     <xsl:if test="$typeOf008!=''">
465     <span class="results_material_type">
466         <span class="label">Material type: </span>
467         <xsl:choose>
468             <xsl:when test="$leader19='a'"> Set</xsl:when>
470             <xsl:when test="$leader6='a'">
471                 <xsl:choose>
472                     <xsl:when test="$leader7='c' or $leader7='d' or $leader7='m'"
473                         > Text</xsl:when>
475                     <xsl:when test="$leader7='i' or $leader7='s'"><img src=
476                         "/opac-tmpl/lib/famfamfam/SE.png" alt="Continuing resource" class
477                             ="materialtype mt_icon_SE"/> Continuing resource</xsl:when>
478                     <xsl:when test="$leader7='a' or $leader7='b'"><img src=
479                         "/opac-tmpl/lib/famfamfam/AR.png" alt="Article" class=
480                             "materialtype mt_icon_AR"/> Article</xsl:when>
481                 </xsl:choose>
482             </xsl:when>
483             <xsl:when test="$leader6='t'"> Text</xsl:when>
485             <xsl:when test="$leader6='o'"> Kit</xsl:when>
487             <xsl:when test="$leader6='p'">Mixed materials
489             </xsl:when>
490             <xsl:when test="$leader6='m'"> Computer file
492             </xsl:when>
493             <xsl:when test="$leader6='e' or $leader6='f'"><img src=
494                 "/opac-tmpl/lib/famfamfam/MP.png" alt="Map" class="materialtype
495                 mt_icon_MP"/> Map</xsl:when>
496             <xsl:when test="$leader6='g'"> Film</xsl:when>
498             <xsl:when test="$leader6='k'"> Picture</xsl:when>
500             <xsl:when test="$leader6='r'"> Object</xsl:when>
502             <xsl:when test="$leader6='c' or $leader6='d'"><img src=
503                 "/opac-tmpl/lib/famfamfam/PR.png" alt="Score" class="materialtype
504                 mt_icon_PR"/> Score</xsl:when>
505             <xsl:when test="$leader6='i'"> Sound</xsl:when>
507             <xsl:when test="$leader6='j'"> Music</xsl:when>
509         </xsl:choose>
510     </span>
511 </xsl:if>
512 <xsl:if test="string-length(normalize-space($physicalDescription))">
513     <span class="results_format">
514         <span class="label">; Format: </span><xsl:copy-of select=
515             "$physicalDescription"></xsl:copy-of>
516     </span>
517 </xsl:if>
518
519     <xsl:if test="$controlField008-21 or $controlField008-22 or $controlField008-24
520         or $controlField008-26 or $controlField008-29 or $controlField008-34 or
521         $controlField008-33 or $controlField008-30-31 or $controlField008-33">
522
523 <xsl:if test="$typeOf008='CR'">

```

```

499 <span class="results_typeofcontinuing">
500 <xsl:if test="$controlField008-21 and $controlField008-21 !='|' and
    $controlField008-21 !=' '">
501 <span class="label">; Type of continuing resource: </span>
502 </xsl:if>
503 <xsl:choose>
504 <xsl:when test="$controlField008-21='d'">
505 
506 </xsl:when>
507 <xsl:when test="$controlField008-21='l'">
508 <b>loose-leaf</b>
509 </xsl:when>
510 <xsl:when test="$controlField008-21='m'">
511 <b>series</b>
512 </xsl:when>
513 <xsl:when test="$controlField008-21='n'">
514 <b>newspaper</b>
515 </xsl:when>
516 <xsl:when test="$controlField008-21='p'">
517 <b>periodical</b>
518 </xsl:when>
519 <xsl:when test="$controlField008-21='w'">
520 
521 </xsl:when>
522 </xsl:choose>
523 </span>
524 </xsl:if>
525 <xsl:if test="$typeOf008='BK' or $typeOf008='CR'">
526 <xsl:if test="contains($controlField008-24,'abcdefghijklmnopqrstvwxyz)'">
527 <span class="results_natureofcontents">
528 <span class="label">; Nature of contents: </span>
529 <xsl:choose>
530 <xsl:when test="contains($controlField008-24,'a)'">
531 <b>abstract or summary</b>
532 </xsl:when>
533 <xsl:when test="contains($controlField008-24,'b)'">
534 <b>bibliography</b>
535 
536 </xsl:when>
537 <xsl:when test="contains($controlField008-24,'c)'">
538 <b>catalog</b>
539 </xsl:when>
540 <xsl:when test="contains($controlField008-24,'d)'">
541 <b>dictionary</b>
542 </xsl:when>
543 <xsl:when test="contains($controlField008-24,'e)'">
544 <b>encyclopedia</b>
545 </xsl:when>
546 <xsl:when test="contains($controlField008-24,'f)'">
547 <b>handbook</b>
548 </xsl:when>
549 <xsl:when test="contains($controlField008-24,'g)'">
550 <b>legal article</b>
551 </xsl:when>
552 <xsl:when test="contains($controlField008-24,'i)'">
553 <b>index</b>
554 </xsl:when>
555 <xsl:when test="contains($controlField008-24,'k)'">
556 <b>discography</b>
557 </xsl:when>
558 <xsl:when test="contains($controlField008-24,'l)'">
559 <b>legislation</b>
560 </xsl:when>
561 <xsl:when test="contains($controlField008-24,'m)'">

```

```

562         theses
563     </xsl:when>
564     <xsl:when test="contains($controlField008-24,'n')">
565         survey of literature
566     </xsl:when>
567     <xsl:when test="contains($controlField008-24,'o')">
568         review
569     </xsl:when>
570     <xsl:when test="contains($controlField008-24,'p')">
571         programmed text
572     </xsl:when>
573     <xsl:when test="contains($controlField008-24,'q')">
574         filmography
575     </xsl:when>
576     <xsl:when test="contains($controlField008-24,'r')">
577         directory
578     </xsl:when>
579     <xsl:when test="contains($controlField008-24,'s')">
580         statistics
581     </xsl:when>
582     <xsl:when test="contains($controlField008-24,'t')">
583         
585     </xsl:when>
586     <xsl:when test="contains($controlField008-24,'v')">
587         legal case and case notes
588     </xsl:when>
589     <xsl:when test="contains($controlField008-24,'w')">
590         law report or digest
591     </xsl:when>
592     <xsl:when test="contains($controlField008-24,'z')">
593         treaty
594     </xsl:when>
595 </xsl:choose>
596 <xsl:choose>
597     <xsl:when test="$controlField008-29='1'">
598         conference publication
599     </xsl:when>
600 </xsl:choose>
601 </span>
602 </xsl:if>
603 </xsl:if>
604 <xsl:if test="$typeOf008='CF'">
605 <span class="results_typeofcomp">
606     <xsl:if test="$controlField008-26='a' or $controlField008-26='e' or
607         $controlField008-26='f' or $controlField008-26='g'">
608     <span class="label">; Type of computer file: </span>
609     </xsl:if>
610     <xsl:choose>
611         <xsl:when test="$controlField008-26='a'">
612             numeric data
613         </xsl:when>
614         <xsl:when test="$controlField008-26='e'">
615             
617         </xsl:when>
618         <xsl:when test="$controlField008-26='f'">
619             
621         </xsl:when>
622         <xsl:when test="$controlField008-26='g'">
623             
625         </xsl:when>
626     </xsl:choose>
627 </span>
628 </xsl:if>

```

```

624 <xsl:if test="$typeOf008='BK'">
625 <span class="results_contents_literary">
626 <xsl:if test="(substring($controlField008,25,1)='j') or
(substring($controlField008,25,1)='1') or ($controlField008-34='a' or
$controlField008-34='b' or $controlField008-34='c' or
$controlField008-34='d')">
627 <span class="label">; Nature of contents: </span>
628 </xsl:if>
629 <xsl:if test="substring($controlField008,25,1)='j'">
630 patent
631 </xsl:if>
632 <xsl:if test="substring($controlField008,31,1)='1'">
633 festschrift
634 </xsl:if>
635 <xsl:if test="$controlField008-34='a' or $controlField008-34='b' or
$controlField008-34='c' or $controlField008-34='d'">
636 
637 </xsl:if>
638
639 <xsl:if test="$controlField008-33 and $controlField008-33!='|' and
$controlField008-33!='u' and $controlField008-33!=' '">
640 <span class="label">; Literary form: </span>
641 </xsl:if>
642 <xsl:choose>
643 <xsl:when test="$controlField008-33='0'">
644 Not fiction
645 </xsl:when>
646 <xsl:when test="$controlField008-33='1'">
647 Fiction
648 </xsl:when>
649 <xsl:when test="$controlField008-33='d'">
650 Dramas
651 </xsl:when>
652 <xsl:when test="$controlField008-33='e'">
653 Essays
654 </xsl:when>
655 <xsl:when test="$controlField008-33='f'">
656 Novels
657 </xsl:when>
658 <xsl:when test="$controlField008-33='h'">
659 Humor, satires, etc.
660 </xsl:when>
661 <xsl:when test="$controlField008-33='i'">
662 Letters
663 </xsl:when>
664 <xsl:when test="$controlField008-33='j'">
665 Short stories
666 </xsl:when>
667 <xsl:when test="$controlField008-33='m'">
668 Mixed forms
669 </xsl:when>
670 <xsl:when test="$controlField008-33='p'">
671 Poetry
672 </xsl:when>
673 <xsl:when test="$controlField008-33='s'">
674 Speeches
675 </xsl:when>
676 </xsl:choose>
677 </span>
678 </xsl:if>
679 <xsl:if test="$typeOf008='MU' and $controlField008-30-31 and
$controlField008-30-31!='||' and $controlField008-30-31!=' '">
680 <span class="results_literaryform">
681 <span class="label">; Literary form: </span> <!-- Literary text for sound
recordings -->
682 <xsl:if test="contains($controlField008-30-31,'b')">

```

```

683         biography
684     </xsl:if>
685     <xsl:if test="contains($controlField008-30-31,'c')">
686         conference publication
687     </xsl:if>
688     <xsl:if test="contains($controlField008-30-31,'d')">
689         drama
690     </xsl:if>
691     <xsl:if test="contains($controlField008-30-31,'e')">
692         essay
693     </xsl:if>
694     <xsl:if test="contains($controlField008-30-31,'f')">
695         fiction
696     </xsl:if>
697     <xsl:if test="contains($controlField008-30-31,'o')">
698         folktale
699     </xsl:if>
700     <xsl:if test="contains($controlField008-30-31,'h')">
701         history
702     </xsl:if>
703     <xsl:if test="contains($controlField008-30-31,'k')">
704         humor, satire
705     </xsl:if>
706     <xsl:if test="contains($controlField008-30-31,'m')">
707         memoir
708     </xsl:if>
709     <xsl:if test="contains($controlField008-30-31,'p')">
710         poetry
711     </xsl:if>
712     <xsl:if test="contains($controlField008-30-31,'r')">
713         rehearsal
714     </xsl:if>
715     <xsl:if test="contains($controlField008-30-31,'g')">
716         reporting
717     </xsl:if>
718     <xsl:if test="contains($controlField008-30-31,'s')">
719         sound
720     </xsl:if>
721     <xsl:if test="contains($controlField008-30-31,'l')">
722         speech
723     </xsl:if>
724 </span>
725 </xsl:if>
726 <xsl:if test="$typeOf008='VM'">
727 <span class="results_typeofvisual">
728     <span class="label">; Type of visual material: </span>
729     <xsl:choose>
730         <xsl:when test="$controlField008-33='a'">
731             art original
732         </xsl:when>
733         <xsl:when test="$controlField008-33='b'">
734             kit
735         </xsl:when>
736         <xsl:when test="$controlField008-33='c'">
737             art reproduction
738         </xsl:when>
739         <xsl:when test="$controlField008-33='d'">
740             diorama
741         </xsl:when>
742         <xsl:when test="$controlField008-33='f'">
743             filmstrip
744         </xsl:when>
745         <xsl:when test="$controlField008-33='g'">
746             game
747         </xsl:when>
748         <xsl:when test="$controlField008-33='i'">
749             picture

```

```

750         </xsl:when>
751         <xsl:when test="$controlField008-33='k'">
752             graphic
753         </xsl:when>
754         <xsl:when test="$controlField008-33='l'">
755             technical drawing
756         </xsl:when>
757         <xsl:when test="$controlField008-33='m'">
758             motion picture
759         </xsl:when>
760         <xsl:when test="$controlField008-33='n'">
761             chart
762         </xsl:when>
763         <xsl:when test="$controlField008-33='o'">
764             flash card
765         </xsl:when>
766         <xsl:when test="$controlField008-33='p'">
767             microscope slide
768         </xsl:when>
769         <xsl:when test="$controlField008-33='q' or
marc:controlfield[@tag=007][substring(text(),1,1)='a'][substring(text(),2
,1)='q']">
770             model
771         </xsl:when>
772         <xsl:when test="$controlField008-33='r'">
773             realia
774         </xsl:when>
775         <xsl:when test="$controlField008-33='s'">
776             slide
777         </xsl:when>
778         <xsl:when test="$controlField008-33='t'">
779             transparency
780         </xsl:when>
781         <xsl:when test="$controlField008-33='v'">
782             videorecording
783         </xsl:when>
784         <xsl:when test="$controlField008-33='w'">
785             toy
786         </xsl:when>
787     </xsl:choose>
788 </span>
789 </xsl:if>
790 </xsl:if>
791
792 <xsl:if test="($typeOf008='BK' or $typeOf008='CF' or $typeOf008='MU' or
$typeOf008='VM') and ($controlField008-22='a' or $controlField008-22='b' or
$controlField008-22='c' or $controlField008-22='d' or $controlField008-22='e' or
$controlField008-22='g' or $controlField008-22='j' or $controlField008-22='f')">
793     <span class="results_audience">
794     <span class="label">; Audience: </span>
795     <xsl:choose>
796         <xsl:when test="$controlField008-22='a'">
797             Preschool;
798         </xsl:when>
799         <xsl:when test="$controlField008-22='b'">
800             Primary;
801         </xsl:when>
802         <xsl:when test="$controlField008-22='c'">
803             Pre-adolescent;
804         </xsl:when>
805         <xsl:when test="$controlField008-22='d'">
806             Adolescent;
807         </xsl:when>
808         <xsl:when test="$controlField008-22='e'">
809             Adult;
810         </xsl:when>
811         <xsl:when test="$controlField008-22='g'">

```

```

812         General;
813     </xsl:when>
814     <xsl:when test="$controlField008-22='j'">
815         Juvenile;
816     </xsl:when>
817     <xsl:when test="$controlField008-22='f'">
818         Specialized;
819     </xsl:when>
820 </xsl:choose>
821 </span>
822 </xsl:if>
823 <xsl:text> </xsl:text> <!-- added blank space to fix font display problem, see Bug 3671
-->
824 </div>
825 </xsl:if>
826
827     <xsl:call-template name="show-lang-041"/>
828
829     <!-- Publisher Statement: Alternate Graphic Representation (MARC 880) -->
830     <xsl:if test="$display880">
831         <xsl:call-template name="m880Select">
832             <xsl:with-param name="basetags">260</xsl:with-param>
833             <xsl:with-param name="codes">abcg</xsl:with-param>
834             <xsl:with-param name="class">results_summary_publisher</xsl:with-param>
835             <xsl:with-param name="label">Publication details: </xsl:with-param>
836         </xsl:call-template>
837     </xsl:if>
838
839
840     <!-- PMID -->
841     <!-- Modified for authcat: changed tag 099-->
842     <xsl:if test="marc:datafield[@tag=099]">
843         <div class="results_summary_publisher_number ">
844             <span class="label">PMID: </span>
845             <ul class="resource_list">
846                 <xsl:for-each select="marc:datafield[@tag=099]">
847                     <li>
848                         <xsl:call-template name="subfieldSelect">
849                             <xsl:with-param name="codes">a</xsl:with-param>
850                         </xsl:call-template>
851                     </li>
852                 </xsl:for-each>
853             </ul>
854         </div>
855     </xsl:if>
856
857
858     <!-- Publisher info and RDA related info from tags 260, 264 -->
859     <xsl:choose>
860         <xsl:when test="marc:datafield[@tag=264]">
861             <xsl:call-template name="showRDAtag264"/>
862         </xsl:when>
863         <xsl:when test="marc:datafield[@tag=260]">
864     <!-- Modified for authcat: label changed to year-->
865         <div class="results_summary_publisher"><span class="label">Year: </span>
866             <xsl:for-each select="marc:datafield[@tag=260]">
867     <!--
868                 <xsl:if test="marc:subfield[@code='a']">
869                     <span class="publisher_place" property="location">
870                         <xsl:call-template name="subfieldSelect">
871                             <xsl:with-param name="codes">a</xsl:with-param>
872                         </xsl:call-template>
873                     </span>
874                 </xsl:if>
875                 <xsl:text> </xsl:text>
876                 <xsl:if test="marc:subfield[@code='b']">
877                     <span property="name" class="publisher_name">

```



```

878         <xsl:call-template name="subfieldSelect">
879             <xsl:with-param name="codes">b</xsl:with-param>
880         </xsl:call-template>
881     </span>
882 </xsl:if>
883 <xsl:text> </xsl:text>
884 -->
885     <span property="datePublished" class="publisher_date">
886         <xsl:call-template name="chopPunctuation">
887             <xsl:with-param name="chopString">
888                 <xsl:call-template name="subfieldSelect">
889                     <xsl:with-param name="codes">c</xsl:with-param>
890                 </xsl:call-template>
891             </xsl:with-param>
892         </xsl:call-template>
893     </span>
894 <!-- semi-colon taken out -->
895     <xsl:choose><xsl:when test="position()=last()"
896     ><xsl:text></xsl:text></xsl:when><xsl:otherwise><xsl:text></xsl:text>
897     </xsl:otherwise></xsl:choose>
898 </xsl:for-each>
899 <xsl:if test="marc:datafield[@tag=264]">
900     <xsl:text>; </xsl:text>
901     <xsl:call-template name="showRDAtag264"/>
902 </xsl:if>
903 </div>
904 </xsl:when>
905 </xsl:choose>
906 <!-- citation -->
907 <!-- Modified for authcat: new tag -->
908 <xsl:if test="marc:datafield[@tag=251]">
909     <div class="results_summary ">
910         <span class="label">Citation: </span>
911         <ul class="resource_list">
912             <xsl:for-each select="marc:datafield[@tag=251]">
913                 <li>
914                     <xsl:call-template name="subfieldSelect">
915                         <xsl:with-param name="codes">a</xsl:with-param>
916                     </xsl:call-template>
917                 </li>
918             </xsl:for-each>
919         </ul>
920     </div>
921 </xsl:if>
922 <!-- institution -->
923 <!-- Modified for authcat: changed tag -->
924 <xsl:if test="marc:datafield[@tag=651]">
925     <div class="results_summary ">
926         <span class="label">Institution: </span>
927         <ul class="resource_list">
928             <xsl:for-each select="marc:datafield[@tag=651]">
929                 <li>
930                     <xsl:call-template name="subfieldSelect">
931                         <xsl:with-param name="codes">a</xsl:with-param>
932                     </xsl:call-template>
933                 </li>
934             </xsl:for-each>
935         </ul>
936     </div>
937 </xsl:if>
938 <!-- department -->
939 <!-- Modified for authcat: changed tag -->
940 <xsl:if test="marc:datafield[@tag=656]">
941     <div class="results_summary ">

```

```

943     <span class="label">Department: </span>
944     <ul class="resource_list">
945         <xsl:for-each select="marc:datafield[@tag=656]">
946             <li>
947                 <xsl:call-template name="subfieldSelect">
948                     <xsl:with-param name="codes">a</xsl:with-param>
949                 </xsl:call-template>
950             </li>
951         </xsl:for-each>
952     </ul>
953 </div>
954 </xsl:if>
955
956
957 <!-- Publication -->
958 <!-- Modified for authcat: changed tag -->
959 <xsl:if test="marc:datafield[@tag=657]">
960     <div class="results_summary ">
961         <span class="label">Medline publication type: </span>
962         <ul class="resource_list">
963             <xsl:for-each select="marc:datafield[@tag=657]">
964                 <li>
965                     <xsl:call-template name="subfieldSelect">
966                         <xsl:with-param name="codes">a</xsl:with-param>
967                     </xsl:call-template>
968                 </li>
969             </xsl:for-each>
970         </ul>
971     </div>
972 </xsl:if>
973
974
975 <!-- Article type -->
976 <!-- Modified for authcat: changed tag -->
977 <xsl:if test="marc:datafield[@tag=942]">
978     <div class="results_summary ">
979         <span class="label">Form of publication: </span>
980         <ul class="resource_list">
981             <xsl:for-each select="marc:datafield[@tag=942]">
982                 <li>
983                     <xsl:call-template name="subfieldSelect">
984                         <xsl:with-param name="codes">c</xsl:with-param>
985                     </xsl:call-template>
986                 </li>
987             </xsl:for-each>
988         </ul>
989     </div>
990 </xsl:if>
991
992
993 <!-- AU -->
994 <!-- Modified for authcat: changed tag -->
995 <xsl:if test="marc:datafield[@tag=790]">
996     <div class="results_summary ">
997         <span class="label">All authors: </span>
998         <ul class="resource_list">
999             <xsl:for-each select="marc:datafield[@tag=790]">
1000                 <li>
1001                     <xsl:call-template name="subfieldSelect">
1002                         <xsl:with-param name="codes">a</xsl:with-param>
1003                     </xsl:call-template>
1004                 </li>
1005             </xsl:for-each>
1006         </ul>
1007     </div>
1008 </xsl:if>
1009

```

```

1010 <!-- Dissertation note -->
1011 <xsl:if test="marc:datafield[@tag=502]">
1012   <div class="results_summary_diss_note">
1013     <span class="label">Dissertation note: </span>
1014     <xsl:for-each select="marc:datafield[@tag=502]">
1015       <xsl:call-template name="subfieldSelect">
1016         <xsl:with-param name="codes">abcdgo</xsl:with-param>
1017       </xsl:call-template>
1018     </xsl:for-each>
1019     <xsl:choose><xsl:when test="position()=last()"
1020       ><xsl:text></xsl:text></xsl:when><xsl:otherwise><xsl:text>
1021       </xsl:text></xsl:otherwise></xsl:choose>
1022   </div>
1023 </xsl:if>
1024
1025 <!-- Other Title Statement: Alternate Graphic Representation (MARC 880) -->
1026 <xsl:if test="$display880">
1027   <xsl:call-template name="m880Select">
1028     <!-- Modified for authcat: -->
1029     <xsl:with-param name="basetags">099</xsl:with-param>
1030     <xsl:with-param name="codes">ab</xsl:with-param>
1031     <xsl:with-param name="class">results_summary_other_title</xsl:with-param>
1032     <xsl:with-param name="label">PMID: </xsl:with-param>
1033   </xsl:call-template>
1034 </xsl:if>
1035
1036 <xsl:if test="marc:datafield[@tag=246]">
1037   <div class="results_summary_other_title">
1038     <span class="label">Other title: </span>
1039     <ul class="resource_list">
1040       <xsl:for-each select="marc:datafield[@tag=246]">
1041         <li>
1042           <xsl:call-template name="subfieldSelect">
1043             <xsl:with-param name="codes">ab</xsl:with-param>
1044           </xsl:call-template>
1045         </li>
1046       </xsl:for-each>
1047     </ul>
1048   </div>
1049 </xsl:if>
1050
1051 <xsl:if test="marc:datafield[@tag=242]">
1052   <div class="results_summary_translated_title">
1053     <span class="label">Title translated: </span>
1054     <xsl:for-each select="marc:datafield[@tag=242]">
1055       <xsl:call-template name="subfieldSelect">
1056         <xsl:with-param name="codes">abh</xsl:with-param>
1057       </xsl:call-template>
1058       <xsl:choose><xsl:when test="position()=last()"><xsl:text>.</xsl:text>.</xsl:when>
1059       <xsl:otherwise><xsl:text>;</xsl:text>;</xsl:otherwise></xsl:choose>
1060     </xsl:for-each>
1061   </div>
1062 </xsl:if>
1063
1064 <xsl:if test="marc:datafield[@tag=856]">
1065   <!-- Modified for authcat: DOI -->
1066   <div class="results_summary_online_resources">
1067     <span class="label">DOI: </span>
1068     <ul class="resource_list">
1069       <xsl:for-each select="marc:datafield[@tag=856]">
1070         <xsl:variable name="SubqText"><xsl:value-of select="
1071           marc:subfield[@code='q']"/></xsl:variable>
1072         <xsl:if test="$OPACURLOpenInNewWindow='0'">
1073           <li><a>
1074             <xsl:choose>
1075               <xsl:when test="$OPACTrackClicks='track'">
1076                 <xsl:attribute name="href"/>/cgi-bin/koha/tracklinks.pl?uri=
1077                 <xsl:value-of select="str:encode-uri(marc:subfield[@code='u']),

```

```

1071         true()"/>&amp;biblionenumber=<xsl:value-of select="$biblionumber"/>
1072         </xsl:attribute>
1073     </xsl:when>
        <xsl:when test="$OPACTrackClicks='anonymous'">
1074         <xsl:attribute name="href"/>/cgi-bin/koha/tracklinks.pl?uri=
1075         <xsl:value-of select="str:encode-uri(marc:subfield[@code='u'],
1076         true()"/>&amp;biblionenumber=<xsl:value-of select="$biblionumber"/>
1077         </xsl:attribute>
1078     </xsl:when>
1079     <xsl:otherwise>
1080         <xsl:attribute name="href">
1081             <xsl:if test="not(contains(marc:subfield[@code='u'],'://'))">
1082                 <xsl:choose>
1083                     <xsl:when test="@ind1=7">
1084                         <xsl:value-of select="marc:subfield[@code='2']"/>
1085                         <xsl:text>://</xsl:text>
1086                     </xsl:when>
1087                     <xsl:when test="@ind1=1">
1088                         <xsl:text>ftp://</xsl:text>
1089                     </xsl:when>
1090                     <xsl:otherwise>
1091                         <xsl:text>http://</xsl:text>
1092                     </xsl:otherwise>
1093                 </xsl:choose>
1094             </xsl:if>
1095             <xsl:value-of select="marc:subfield[@code='u']"/>
1096         </xsl:attribute>
1097     </xsl:otherwise>
1098     </xsl:choose>
        <xsl:choose>
1099         <xsl:when test="($Show856uAsImage='Results' or
1100         $Show856uAsImage='Both') and
1101         (substring($SubqText,1,6)='image/' or
1102         $SubqText='img' or $SubqText='bmp' or
1103         $SubqText='cod' or $SubqText='gif' or
1104         $SubqText='ief' or $SubqText='jpe' or
1105         $SubqText='jpeg' or $SubqText='jpg' or
1106         $SubqText='jfif' or $SubqText='png' or
1107         $SubqText='svg' or $SubqText='tif' or
1108         $SubqText='tiff' or $SubqText='ras' or
1109         $SubqText='cmx' or $SubqText='ico' or
1110         $SubqText='pnm' or $SubqText='pbm' or
1111         $SubqText='pgm' or $SubqText='ppm' or
1112         $SubqText='rgb' or $SubqText='xbm' or
1113         $SubqText='xpm' or $SubqText='xwd')">
1114             <xsl:element name="img"><xsl:attribute name="src"
1115             ><xsl:value-of select="marc:subfield[@code='u']"
1116             /></xsl:attribute><xsl:attribute name="alt"
1117             ><xsl:value-of select="marc:subfield[@code='y']"
1118             /></xsl:attribute><xsl:attribute name="style"
1119             >
1120                 height:100px;
1121             </xsl:attribute></xsl:element><xsl:text></xsl:tex
1122             t>
1123         </xsl:when>
1124         <xsl:when test="marc:subfield[@code='y' or @code='3'
1125         or @code='z']">
1126             <xsl:call-template name="subfieldSelect">
1127                 <xsl:with-param name="codes">y3z</xsl:with-param>
1128             </xsl:call-template>
1129         </xsl:when>
1130         <xsl:when test="not(marc:subfield[@code='y']) and
1131         not(marc:subfield[@code='3']) and
1132         not(marc:subfield[@code='z'])">
1133             <xsl:choose>
1134                 <xsl:when test="$URLLinkText!=''">
1135                     <xsl:value-of select="$URLLinkText"/>
1136                 </xsl:when>

```

```

1108 <xsl:otherwise>
1109 <xsl:text>Click here to access online</xsl:text>
1110 </xsl:otherwise>
1111 </xsl:choose>
1112 </xsl:when>
1113 </xsl:choose>
1114 </a></li>
1115 </xsl:if>
1116 <xsl:if test="$OPACURLOpenInNewWindow='1'">
1117 <li><a target='_blank'>
1118 <xsl:choose>
1119 <xsl:when test="$OPACTrackClicks='track'">
1120 <xsl:attribute name="href"/>/cgi-bin/koha/tracklinks.pl?uri=
<xsl:value-of select="str:encode-uri(marc:subfield[@code='u'],
true())"/>&amp;biblionumber=<xsl:value-of select="$biblionumber"/>
</xsl:attribute>
1121 </xsl:when>
1122 <xsl:when test="$OPACTrackClicks='anonymous'">
1123 <xsl:attribute name="href"/>/cgi-bin/koha/tracklinks.pl?uri=
<xsl:value-of select="str:encode-uri(marc:subfield[@code='u'],
true())"/>&amp;biblionumber=<xsl:value-of select="$biblionumber"/>
</xsl:attribute>
1124 </xsl:when>
1125 <xsl:otherwise>
1126 <xsl:attribute name="href"><xsl:value-of select=
"marc:subfield[@code='u']"/></xsl:attribute>
1127 </xsl:otherwise>
1128 </xsl:choose>
1129 <xsl:choose>
1130 <xsl:when test="($Show856uAsImage='Results' or
$Show856uAsImage='Both') and ($SubqText='img' or
$SubqText='bmp' or $SubqText='cod' or
$SubqText='gif' or $SubqText='ief' or
$SubqText='jpe' or $SubqText='jpeg' or
$SubqText='jpg' or $SubqText='jfif' or
$SubqText='png' or $SubqText='svg' or
$SubqText='tif' or $SubqText='tiff' or
$SubqText='ras' or $SubqText='cmx' or
$SubqText='ico' or $SubqText='pnm' or
$SubqText='pbm' or $SubqText='pgm' or
$SubqText='ppm' or $SubqText='rgb' or
$SubqText='xbm' or $SubqText='xpm' or
$SubqText='xwd')">
1131 <xsl:element name="img"><xsl:attribute name="src"
><xsl:value-of select="marc:subfield[@code='u']"
/></xsl:attribute><xsl:attribute name="alt"
><xsl:value-of select="marc:subfield[@code='y']"
/></xsl:attribute><xsl:attribute name="style">
height:100px
</xsl:attribute></xsl:element><xsl:text></xsl:tex
t>
1132 </xsl:when>
1133 <xsl:when test="marc:subfield[@code='y' or @code='3'
or @code='z']">
1134 <xsl:call-template name="subfieldSelect">
1135 <xsl:with-param name="codes">y3z</xsl:with-param>
1136 </xsl:call-template>
1137 </xsl:when>
1138 <xsl:when test="not(marc:subfield[@code='y']) and
not(marc:subfield[@code='3']) and
not(marc:subfield[@code='z'])">
1139 <xsl:choose>
1140 <xsl:when test="$URLLinkText!=''">
1141 <xsl:value-of select="$URLLinkText"/>
1142 </xsl:when>
1143 <xsl:otherwise>
1144 <xsl:text>Click here to access online</xsl:text>

```

```

1145         </xsl:otherwise>
1146     </xsl:choose>
1147         </xsl:when>
1148     </xsl:choose>
1149     </a></li>
1150     </xsl:if>
1151 </xsl:for-each>
1152 </ul>
1153 </div>
1154 </xsl:if>
1155
1156
1157     <!-- Local accession number -->
1158 <!-- Modified for authcat: changed tag -->
1159     <xsl:if test="marc:datafield[@tag=098]">
1160         <div class="results_summary ">
1161             <span class="label">Local accession number: </span>
1162             <ul class="resource_list">
1163                 <xsl:for-each select="marc:datafield[@tag=098]">
1164                     <li>
1165                         <xsl:call-template name="subfieldSelect">
1166                             <xsl:with-param name="codes">a</xsl:with-param>
1167                         </xsl:call-template>
1168                     </li>
1169                 </xsl:for-each>
1170             </ul>
1171         </div>
1172     </xsl:if>
1173
1174
1175
1176     <!-- Availability line -->
1177 <!-- Modified for authcat: availability removed -->
1178
1179     <!-- End of Availability line -->
1180
1181 </xsl:template>
1182
1183 <xsl:template name="nameABCQ">
1184     <xsl:call-template name="chopPunctuation">
1185         <xsl:with-param name="chopString">
1186             <xsl:call-template name="subfieldSelect">
1187                 <xsl:with-param name="codes">abcq</xsl:with-param>
1188             </xsl:call-template>
1189         </xsl:with-param>
1190         <xsl:with-param name="punctuation">
1191             <xsl:text>.,;/ </xsl:text>
1192         </xsl:with-param>
1193     </xsl:call-template>
1194 </xsl:template>
1195
1196 <xsl:template name="nameABCDN">
1197     <xsl:call-template name="chopPunctuation">
1198         <xsl:with-param name="chopString">
1199             <xsl:call-template name="subfieldSelect">
1200                 <xsl:with-param name="codes">abcdn</xsl:with-param>
1201             </xsl:call-template>
1202         </xsl:with-param>
1203         <xsl:with-param name="punctuation">
1204             <xsl:text>.,;/ </xsl:text>
1205         </xsl:with-param>
1206     </xsl:call-template>
1207 </xsl:template>
1208
1209 <xsl:template name="nameACDEQ">
1210     <xsl:call-template name="subfieldSelect">
1211         <xsl:with-param name="codes">acdeq</xsl:with-param>

```

```

1212         </xsl:call-template>
1213     </xsl:template>
1214
1215     <xsl:template name="nameDate">
1216         <xsl:for-each select="marc:subfield[@code='d']">
1217             <xsl:call-template name="chopPunctuation">
1218                 <xsl:with-param name="chopString" select="."/>
1219             </xsl:call-template>
1220         </xsl:for-each>
1221     </xsl:template>
1222
1223     <xsl:template name="role">
1224         <xsl:for-each select="marc:subfield[@code='e']">
1225             <xsl:value-of select="."/>
1226         </xsl:for-each>
1227         <xsl:for-each select="marc:subfield[@code='4']">
1228             <xsl:value-of select="."/>
1229         </xsl:for-each>
1230     </xsl:template>
1231
1232     <xsl:template name="specialSubfieldSelect">
1233         <xsl:param name="anyCodes"/>
1234         <xsl:param name="axis"/>
1235         <xsl:param name="beforeCodes"/>
1236         <xsl:param name="afterCodes"/>
1237         <xsl:variable name="str">
1238             <xsl:for-each select="marc:subfield">
1239                 <xsl:if test="contains($anyCodes, @code) or
1240                     (contains($beforeCodes,@code) and
1241                     following-sibling::marc:subfield[@code=$axis]) or
1242                     (contains($afterCodes,@code) and
1243                     preceding-sibling::marc:subfield[@code=$axis])">
1244                     <xsl:value-of select="text()"/>
1245                     <xsl:text> </xsl:text>
1246                 </xsl:if>
1247             </xsl:for-each>
1248         </xsl:variable>
1249         <xsl:value-of select="substring($str,1,string-length($str)-1)"/>
1250     </xsl:template>
1251
1252     <xsl:template name="subtitle">
1253         <xsl:if test="marc:subfield[@code='b']">
1254             <xsl:call-template name="chopPunctuation">
1255                 <xsl:with-param name="chopString">
1256                     <xsl:value-of select="marc:subfield[@code='b']"/>
1257                 </xsl:with-param>
1258                 <!--<xsl:call-template name="subfieldSelect">
1259                     <xsl:with-param name="codes">b</xsl:with-param>
1260                 </xsl:call-template-->
1261             </xsl:call-template>
1262         </xsl:if>
1263     </xsl:template>
1264
1265     <xsl:template name="chopBrackets">
1266         <xsl:param name="chopString"></xsl:param>
1267         <xsl:variable name="string">
1268             <xsl:call-template name="chopPunctuation">
1269                 <xsl:with-param name="chopString" select="$chopString"></xsl:with-param>
1270             </xsl:call-template>
1271         </xsl:variable>
1272         <xsl:if test="substring($string, 1,1)='['">
1273             <xsl:value-of select="substring($string,2, string-length($string)-2) "
1274             ></xsl:value-of>
1275         </xsl:if>
1276         <xsl:if test="substring($string, 1,1)!='['">
1277             <xsl:value-of select="$string"></xsl:value-of>

```



```

1274     </xsl:if>
1275 </xsl:template>
1276
1277 <xsl:template name="listCallNumbers">
1278     <xsl:param name="items" />
1279     <xsl:param name="max" />
1280     <xsl:param name="status_text" />
1281     <xsl:param name="class_block" />
1282     <xsl:param name="class_status" />
1283     <xsl:param name="OPACItemLocation" />
1284     <xsl:if test="count($items)>0">
1285         <span><xsl:attribute name="class"><xsl:value-of select="$class_block" />
1286             </xsl:attribute>
1287             <span>
1288                 <xsl:attribute name="class"><xsl:value-of select="$class_status" />
1289                 </xsl:attribute>
1290                 <xsl:value-of select="$status_text" />
1291             </span>
1292             <xsl:text> (</xsl:text>
1293             <xsl:value-of select="count($items)" />
1294             <xsl:text>)</xsl:text>
1295             <xsl:if test="$max>0 and count($items[items:itemcallnumber!=''])>0 and
1296             $OPACItemLocation!='library'">
1297                 <span class="CallNumberAndLabel">
1298                     <span class="LabelCallNumber">
1299                         <xsl:if test="$OPACItemLocation='callnum'">Call number: </xsl:if>
1300                         <xsl:if test="$OPACItemLocation='ccode'">Collection, call
1301                         number: </xsl:if>
1302                         <xsl:if test="$OPACItemLocation='location'">Location, call
1303                         number: </xsl:if>
1304                     </span>
1305                     <span class="CallNumber">
1306                         <xsl:for-each select="$items[items:itemcallnumber!=''] [position()
1307                         &lt;= $max]">
1308                             <xsl:if test="$OPACItemLocation='location'">
1309                                 <strong><xsl:value-of select="concat(items:location,' ')" />
1310                                 </strong>
1311                             </xsl:if>
1312                             <xsl:if test="$OPACItemLocation='ccode'">
1313                                 <strong><xsl:value-of select="concat(items:ccode,' ')" />
1314                                 </strong>
1315                             </xsl:if>
1316                             <xsl:value-of select="items:itemcallnumber" />
1317                             <xsl:if test="position() !=last()">
1318                                 <xsl:text>, </xsl:text>
1319                             </xsl:if>
1320                             <xsl:if test="position()=last() and (count($items)>$max or
1321                             count($items[items:itemcallnumber=''])>0)">
1322                                 <xsl:text>, ..</xsl:text>
1323                             </xsl:if>
1324                         </xsl:for-each>
1325                     </span>
1326                 </span>
1327             </xsl:if>
1328             <xsl:text>. </xsl:text>
1329         </span>
1330     </xsl:if>
1331 </xsl:template>
1332
1333 </xsl:stylesheet>
1334

```