

How do I register "Resources"?

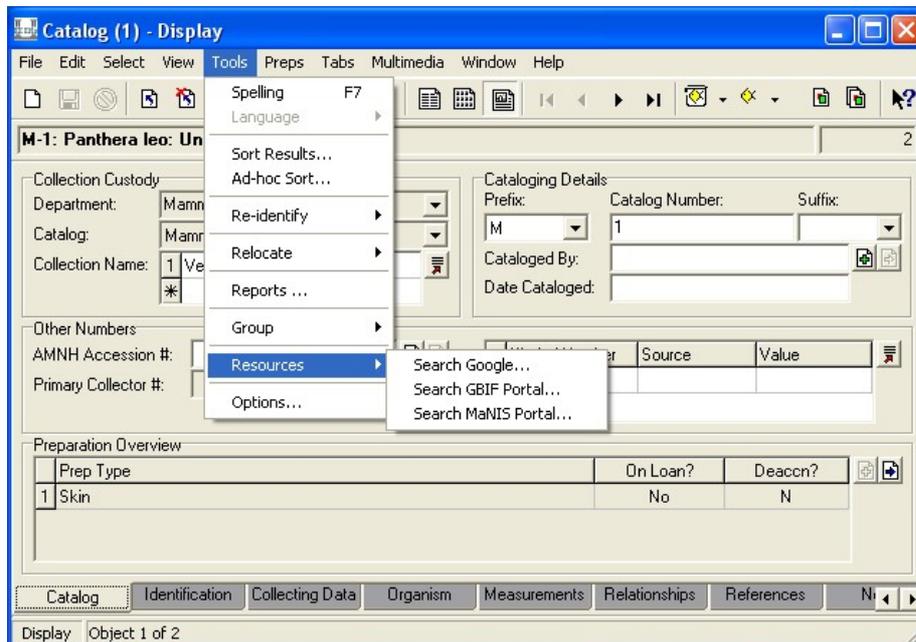
- [Overview](#)
- [Registry entry](#)
- [Resource definition](#)
- [XML reference](#)
- [Examples](#)

Overview

The Resources option is introduced with KE EMu 3.1. A Resource is a string, generally a URL, generated from a Registry entry and pointing to a document, such as an HTML document or search engine. The document could also be a local document (an MS Word file for example) or a document available on a network drive.

The most common use of Resources is to extract data from the current record, insert it into a URL and send the URL to a search engine. For example, a Natural History institution wants to extract the genus and species from a record and search GBIF (Global Biodiversity Information Facility) using that information. The Resource Registry entry allows URLs to be generated dynamically based on data found within the current record.

Users can use the Resource utility to expand a search within KE EMu to resources beyond the scope of their local system. The Resources utility appears under the Tools menu (**Tools>Resources**):



The sub-menu entries (e.g. Search Google) are defined by the Registry entry.

Registry entry

To add an entry to the Resources sub-menu, it is necessary to create a Registry entry that defines:

1. the sub-menu entries; and
2. the resource to display when the sub-menu entry is selected.

The format of the Registry entry is:

```
User | username | Table | table | Resource | title | resource
User | username | Table | Default | Resource | title | resource
Group | groupname | Table | table | Resource | title | resource
Group | groupname | Table | Default | Resource | title | resource
Group | Default | Table | table | Resource | title | resource
Group | Default | Table | Default | Resource | title | resource
```

The entries display in order of precedence from highest in the list to lowest; that is, where the same *title* is given, the entry highest in the list takes precedence. Thus in this example:

```
User | fred | Table | ecatalogue | Resource | Search Google | resource1
Group | Default | Table | Default | Resource | Search Google | resource2
Group | Default | Table | Default | Resource | Search GBIF | resource3
```

when all users, including *fred*, select **Tools>Resources**, the Resources sub-menu will display two entries: **Search Google** and **Search GBIF**. However, when *fred* selects **Tools>Resources>Search Google**, the resource shown will be *resource1*, whereas all other users will see *resource2*.

Resources shown on the sub-menu are sorted alphabetically by title.

Resource definition

The *resource* part of the Registry entry defines a path to the resource to be displayed. In general this path takes the form of a URL, however any other document locator (e.g. file path) may be used. In order to allow data from the current record to be incorporated into a resource path, an XML based parser is used. In other words, you define in XML how the document path is to be constructed. The XML used consists of only six tags, but provides a flexible mechanism for building a document path.

Consider this simple example. The following Resource entry will display the GBIF home page if selected:

```
<resource>
  <location>
    <text>http://www.gbif.org/</text>
  </location>
</resource>
```

The first thing to notice is that the whole entry is enclosed within the `<resource></resource>` tags. The `<location>` tags enclose the `<text>` tags which generate a single document path. A number of `<location>` tags may appear within the one resource.

Each `<location>` tag generates one document path. When processing occurs, each `<location>` section is evaluated, one after another, until a non-empty document path is returned. If processing finishes and no non-empty document path has been returned, the resource menu option is removed from the Resources sub-menu.

The `<text>` tag defines text used to build a document path. In the example above, output from executing the XML would be `http://www.gbif.org/`. If a user selected this resource from the Resources sub-menu, their browser would start up and display the GBIF home page.

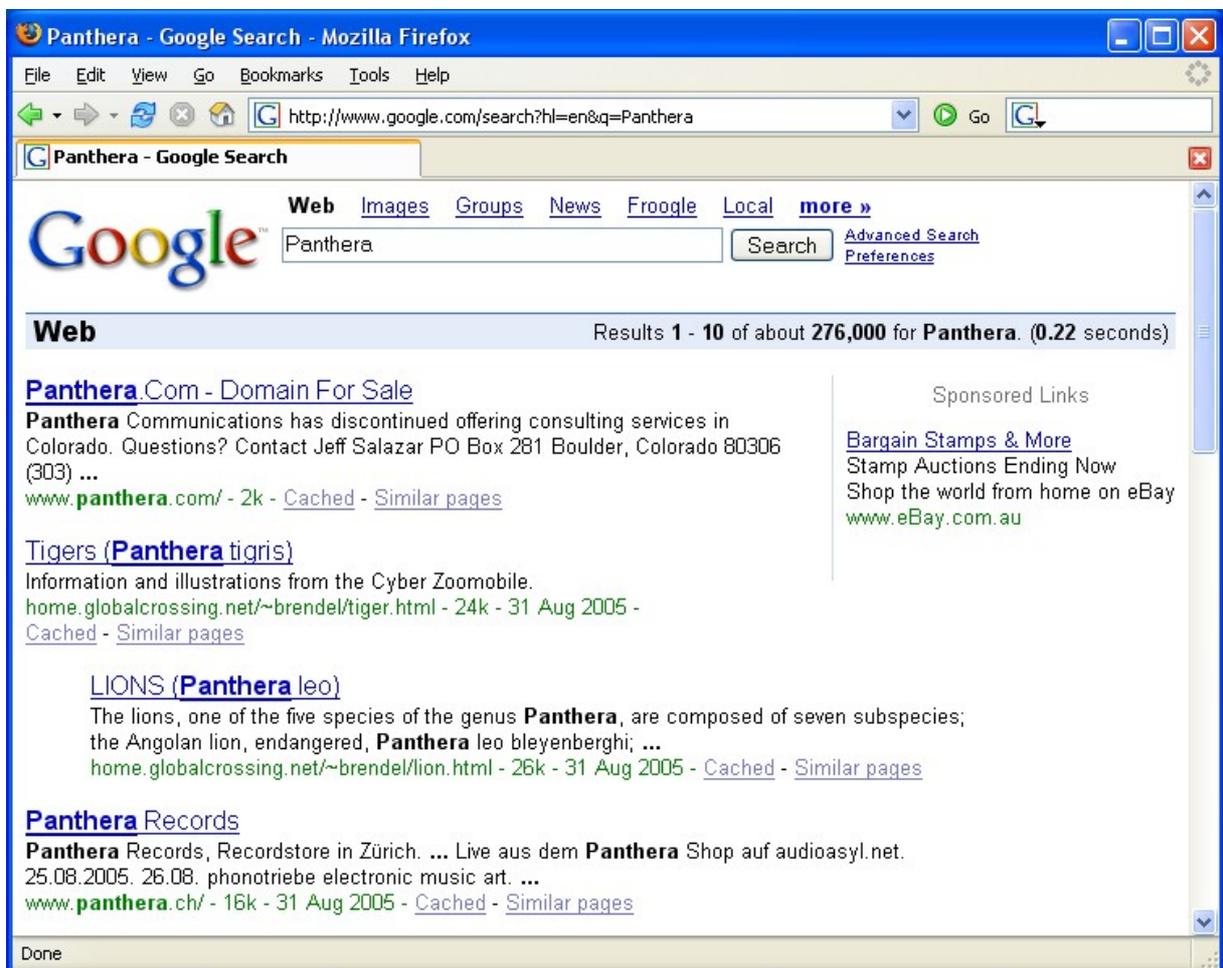
The example above, while useful, is a bit limiting. Ideally we want to include data from the current record. The `<value>` tag allows data to be extracted and added to a document path. The example below adds the value in the *IdeGenus* column as a search term for Google:

```

<resource>
  <location>
    <text>http://www.google.com/search?hl=en&q=</text>
    <value column="IdeGenus" format="url"/>
  </location>
</resource>

```

The `<value>` tag gets the data from the specified *column* attribute (in this case *IdeGenus*) and outputs it as part of the document path. The *format* attribute indicates that the data should be encoded using the URL encoding rules (see [Uniform Resource Locators \(URL\), RFC 1738](#) for details about URL encoding). In effect the encoding ensures that we can send the data via a URL. So if the column *IdeGenus* contains **Panthera**, the output from the above resource would be <http://www.google.com/search?hl=en&q=Panthera>, which would initiate a Google search for the term *Panthera*, displaying the results:



We can combine `<value>` and `<text>` tags to build up quite complex document paths. For example, let's say we wanted to include the genus (*IdeGenus*) and species (*IdeSpecies*) in our Google search. Since we want the terms to appear next to each other we need to enclose the search term in double quotes (URL encoded as %22) with a space between each term (URL encoded as +). The following resource could be used:

```

<resource>
  <location>

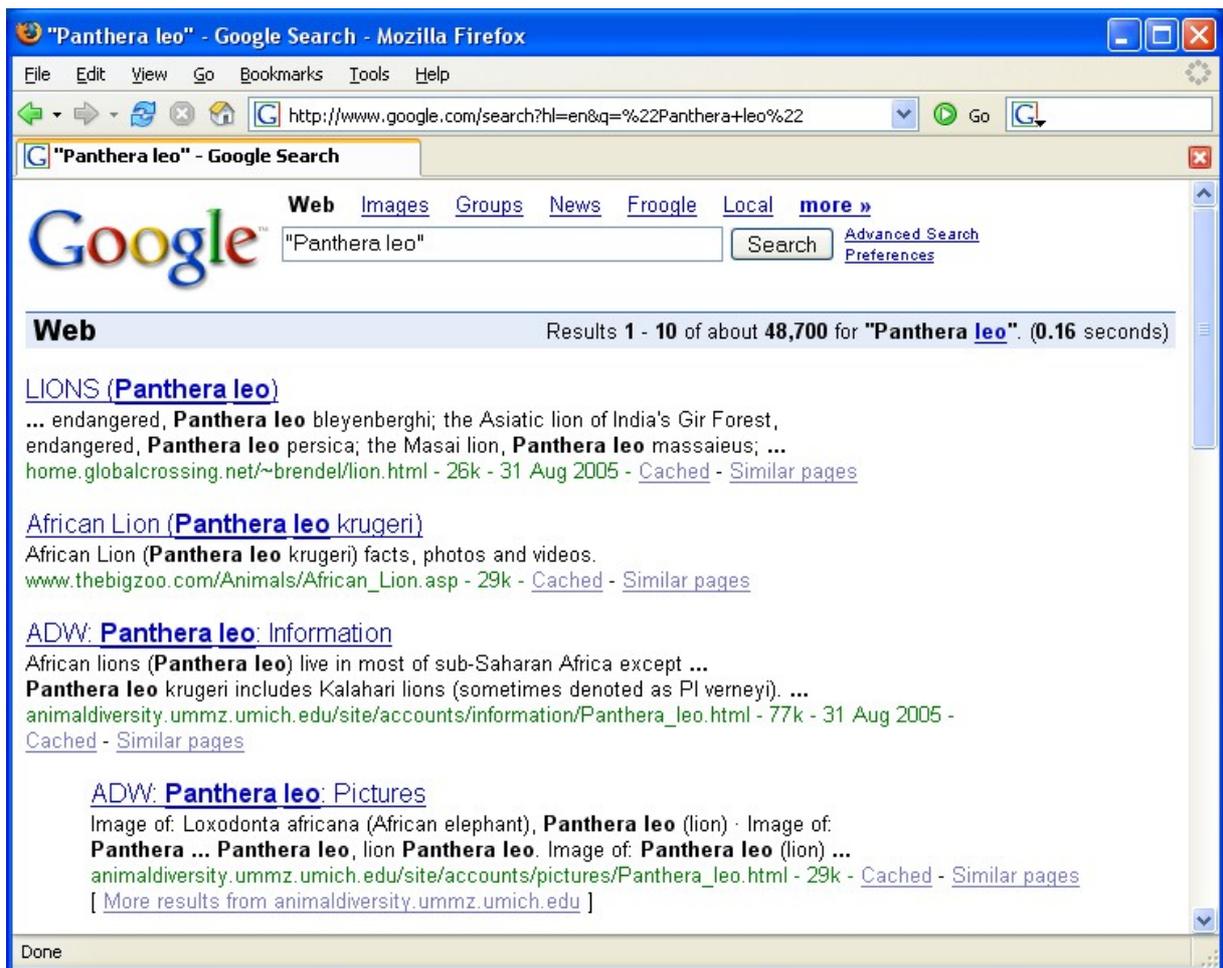
```

```

<text>http://www.google.com/search?hl=en&q=%22</text>
<value column="IdeGenus" format="url" />
<text>+</text>
<value column="IdeSpecies" format="url" />
<text>%22</text>
</location>
</resource>

```

If we have a genus of *Panthera* and a species of *leo*, the document path generated would be <http://www.google.com/search?hl=en&q=%22Panthera+leo%22>, which would initiate a Google search displaying the results:



We can build quite complex document paths using the four tags discussed so far.

The fifth tag, `<if>`, is a conditional tag that provides control over whether a document path should be calculated. In the example above we may only want to perform the Google search if both the genus and species fields contain values. The `<if>` tag allows a field to be tested for either filled/empty or for a given value. If the tag evaluates to *true*, then the inner tags are evaluated, otherwise they are ignored.

In this example we only output the document path if both the genus and species field contain values:

```

<resource>
  <location>

```

```

<if column="IdeGenus" filled="yes">
  <if column="IdeSpecies" filled="yes">
    <text>http://www.google.com/search?hl=en&q=%22</text>
    <value column="IdeGenus" format="url"/>
    <text>+</text>
    <value column="IdeSpecies" format="url"/>
    <text>%22</text>
  </if>
</if>
</location>
</resource>

```

The `<if>` tag takes a *column* used to identify which field is to be tested. The *filled* attribute can be set to *yes* or *no* to indicate whether the field should be filled or empty respectively. So the tag `<if column="IdeGenus" filled="yes">` is true if *IdeGenus* has a value. If it is filled, processing continues with the inner tags (in this case another `<if>` tag), otherwise everything down to the closing `</if>` is ignored.

In the example above, the `<text>` and `<value>` tags are only processed if the genus and species fields contain a value, otherwise an empty document path is generated. If the document path is empty, the Resource is removed from the Resources sub-menu (you can just disable the resource by setting the *visible* attribute on the `<resource>` tag to "yes", that is `<resource visible="yes">`).

The last tag, `<title>`, allows the menu caption to be altered. In particular you may want to include data in the caption. The `<title>` tag can include `<text>`, `<if>` and `<value>` tags. Any output generated by the `<text>` and `<value>` tags, when enclosed in a `<title>` tag, will replace the *title* text defined in the registry.

The following example extends the previous example by adding the genus and species values to the menu caption displayed:

```

<resource>
  <location>
    <if column="IdeGenus" filled="yes">
      <if column="IdeSpecies" filled="yes">
        <text>http://www.google.com/search?hl=en&q=%22</text>
        <value column="IdeGenus" format="url"/>
        <text>+</text>
        <value column="IdeSpecies" format="url"/>
        <text>%22</text>
      </if>
    </if>
    <title>
      <text>Lookup Google for "</text>
      <value column="IdeGenus"/>
      <text> </text>
      <value column="IdeSpecies"/>
      <text>"</text>
    </title>
  </location>
</resource>

```

```

        </title>
      </if>
    </if>
  </location>
</resource>

```

Now that we know all six tags we can use them in combination to build quite sophisticated document paths. As mentioned earlier, the `<location>` tag can repeat, with processing continuing until a non-empty document path is generated. The example below takes advantage of this to generate one of three Google URLs depending on whether the genus and/or species fields are filled:

```

<resource>
  <location>
    <if column="IdeGenus" filled="yes">
      <if column="IdeSpecies" filled="yes">
        <text>http://www.google.com/search?hl=en&q=%22</text>
        <value column="IdeGenus" format="url" />
        <text>+</text>
        <value column="IdeSpecies" format="url" />
        <text>%22</text>
      </if>
    </if>
  </location>
  <location>
    <if column="IdeGenus" filled="yes">
      <text>http://www.google.com/search?hl=en&q=</text>
      <value column="IdeGenus" format="url" />
    </if>
  </location>
  <location>
    <if column="IdeSpecies" filled="yes">
      <text>http://www.google.com/search?hl=en&q=</text>
      <value column="IdeSpecies" format="url" />
    </if>
  </location>
</resource>

```

The first `<location>` section is used if both a genus and species are supplied, the second `<location>` section if only a genus is supplied and the last section if only a species is given. If there is no genus and species, an empty document path results and the Resource is disabled.

XML reference

This section describes each of the five tags and their attributes in detail. It can be used as a reference when designing Resource entries. The reference contains the DTD (Document Type Definition) used to validate Resource XML and then describes each of the five tags, detailing the attributes supported by the tag.

DTD

The following DTD can be used to validate the resource XML.

```
<!ELEMENT resource (location)+>
<!ATTLIST resource
  visible (no|yes) "no"
>
<!ELEMENT location (if|value|text|title)*>
<!ELEMENT if (if|value|text|title)*>
<!ATTLIST if
  column CDATA #REQUIRED
  filled (no|yes) #IMPLIED
  rows CDATA "1"
  value CDATA #IMPLIED
>
<!ELEMENT value EMPTY>
<!ATTLIST value
  column CDATA #REQUIRED
  format (none|url) "none"
  rows CDATA "1"
  separator CDATA " "
>
<!ELEMENT text (#PCDATA)>
<!ELEMENT title (if|value|text)>
```

<resource> tag

The <resource> tag is the outermost tag of a Resource definition. It is used to enclose the definition. A <resource> tag must contain one or more <location> tags.

Attributes

```
  visible="no|yes"
```

The `visible` attribute is used to indicate what should occur if a Resource does not produce a document path. Normally the Resource title is not shown under the Resources sub-menu (that is, the menu entry is not visible). If the `visible` attribute is set to **yes**, the title is shown under the Resources sub-menu, but it is disabled (greyed out). The default case is `visible="no"`.

<location> tag

The <location> tag is used to define a single instance of a document path. A Resource can contain one or more <location> tags. When determining the document path for a given Resource, each <location> section is evaluated one after another, starting from the top, until a non-empty document path is generated. Thus the Boolean relationship between each <location> section is OR. A location element can contain a combination of <if>, <value> and <text> tags.

Attributes

None

<text> tag

The <text> tag is used to output text to build a document path. The text element outputs the characters found between the opening and closing tags. The text is not interpreted in any way (in essence it is treated as CDATA). A text element cannot contain any other tags.

Attributes

None

<value> tag

The <value> tag retrieves the data from the specified *column* and outputs it as part of the document path. The most common use of the value element is to include data as search terms in a query based URL. A value element cannot contain any other tags.

Attributes

```
column="colname"
```

The `column` attribute names the data column from which the data is to be extracted. The name is the column names as used by the KE EMu client. Support is provided for atomic, table and nested table columns. For tables and nested tables the `rows` attribute determines what data is retrieved. In order to provide efficient data access, a cache of column values is maintained. The `column` attribute is mandatory.

```
rows="rows"
```

When the nominated column is a table or nested table, the `rows` attribute determines which values are used. The value can be one of three formats. The first format is just a row number. Row numbers are one based (that is the first row is number one). Thus a tag of:

```
<value column="NamRoles_tab" rows="1"/>
```

means that the returned value will be the first row in the *NamRoles_tab* table.

The second format is a range of rows, e.g. `rows="1-5"`, indicating that the first five rows are to be retrieved from the *NamRoles_tab* table. If there are less than five rows, all available rows are returned.

Finally the keyword `all` may be used to indicate that all rows in a table or nested tab should be retrieved (i.e. `rows="all"`).

Note that nested tables are flattened before rows are extracted. This allows the rows to be treated as a single table of values. The default value for rows is "1".

```
separator="string"
```

When more than one value is extracted from a table, the `separator` attribute is used to indicate how the rows in the table should be joined. The *string* sequence is placed between each row in the table. For example, the tag:

```
<value column=NamRoles_tab rows="all" separator="+"/>
```

will get all values in the *NamRoles_tab* field and join them together with a "+" character between each value (the "+" character is used to encode a space in a URL). The default value for `separator` is " " (a space).

```
format="none|url"
```

Once the value is retrieved, the `format` attribute is used to determine how the data should be encoded. A value of `none` results in the data being output without any encoding taking place. In order to transmit data correctly via a URL, a format of `url` should be used. For more information on URL encoding see [Uniform Resource Locators \(URL\), RFC 1738](#). The default value is `none`.

<if> tag

The `<if>` tag is used to test values. It is the only conditional tag provided for Resource generation. The tag allows you to check whether a column contains a value or not, or whether a certain value appears in a column. If the tag evaluates to true, all inner tags (that is all tags down to the closing `</if>` tag) are evaluated, otherwise they are ignored. An `<if>` tag will generally contain `text`, `values` and other `if` elements.

Attributes

```
column="colname"
```

The `column` attribute names the data column whose data is to be tested. The name is the column name as used by the KE EMu client. Support is provided for atomic, table and nested table columns. For tables and nested tables the `rows` attribute determines what data is retrieved. The `column` attribute is mandatory.

```
rows="rows"
```

When the selected column is a table or nested table, the `rows` attribute determines which values are used. The value can be one of three formats. The first format is a single row number. Row numbers are one based (that is, the first row is number one).

The second format is a range of rows, e.g. `rows="1-5"`, indicating that the first five rows are to be retrieved from the *NamRoles_tab* table. If there are less than five rows, all available rows are returned.

Finally the keyword `all` may be used to indicate that all rows in a table or nested table should be retrieved (i.e. `rows="all"`).

Note that nested tables are flattened before the rows are extracted. This allows the rows to be treated as a single table of values. The default value for rows is "1".

```
filled="no|yes"
```

The `filled` attribute is used to test whether a column contains a value. If the column is a table or nested table, `filled` checks whether the table contains any value. A value of `yes` indicates that the column must contain a value for the condition to be true; a value of `no` specifies that the column should not contain a value to evaluate to true.

```
value="string"
```

The `value` attribute allows you to test for the existence of a value in the data. If the supplied *string* appears as one of the values in the column, the tag evaluates to true, otherwise false. Comparison of the *string* against the data is case insensitive.

<title> tag

The `<title>` tag is used to replace the menu caption displayed. The caption is exchanged with any output generated by `<text>` and `<value>` tags appearing between the start and end `<title>` tags. `<if>` tags can be used to control conditionally what text is generated. If the menu caption is altered by this tag, the position of the menu entry is not changed. Thus, you can use the registry entry *title* to define the menu order (via the alphabetic sort) and then alter the caption shown using this tag.

Attributes

None

Examples

Example 1 - Getty AAT Lookup

In this example we use the *Getty Art and Architecture Thesaurus* to look up terms while we are inserting or editing a record. In order to use the resource the user must enter a value into the first field of the *SubSubject_tab* column. The URL format used by the Getty is:

```
http://www.getty.edu/vow/AATServlet?find=term&page=1&note=
```

In order to use the URL we need to add in the *term* from the data. The following resource XML could be used:

```
<resource visible="yes">
  <location>
    <if column="SubSubject_tab" filled="yes">
      <text>http://www.getty.edu/vow/AATServlet?find=</text>
      <value column="SubSubject_tab" format="url" />
      <text>&page=1&note=</text>
    </if>
  </location>
</resource>
```

The complete registry entry would look like:

```
Group|Default|Table|ecatalogue|Resource|Lookup AAT (Getty)|<resource>
<location> <if column="SubSubject_tab" filled="yes">
<text>http://www.getty.edu/vow/AATServlet?find=</text> <value
column="SubSubject_tab" format="url" /> <text>&page=1&note=</text>
</if> </location> </resource>
```

As you can see the Registry entries can get quite long.

Example 2 - Lookup Google for references to a work of art

This time we perform a Google search for works of art. We use the *Title (TitTitle)* and *Artists (ArtArtists_tab)* field when searching.

The format of a URL for Goggle is:

```
http://www.google.com/search?hl=en&q=terms
```

where *terms* is a series of individual values separated by a "+" character. If a term has multiple words, it must be enclosed in double quotes. In this example we assume that the *Title* must have a value, but the *Artists* are optional. The following resource XML could be used:

```
<resource>
  <location>
    <if column="TitTitle" filled="yes">
      <text>http://www.google.com/search?hl=en&q=%22</text>
      <value column="TitTitle" format="url" />
      <text>%22</text>
      <if column="ArtArtists_tab" filled="yes">
        <text>&q=%22</text>
        <value column="ArtArtists_tab" rows="all" separator=" "
format="url"/>
        <text>%22</text>
      </if>
    </if>
  </location>
</resource>
```

Notice how %22 must be used to encode a double quote in a URL. The complete Registry entry would look like:

```
Group|Default|Table|ecatalogue|Resource|Search Google for
work|<resource> <location> <if column="TitTitle" filled="yes">
<text>http://www.google.com/search?hl=en&q=%22</text> <value
column="TitTitle" format="url" /> <text>%22</text> <if
column="ArtArtists_tab" filled="yes"> <text>&q=%22</text> <value
column="ArtArtists_tab" rows="all" separator=" " format="url"/>
<text>%22</text> </if> </if> </location> </resource>
```

Example 3 - Search GBIF for a given genus and species

GBIF (Global Biodiversity Information Facility) provides a portal to taxonomic information (amongst other things). In this example we use the *Genus (IdeGenus)*, *Species (IdeSpecies)* and *Subspecies (IdeSubspecies)* to access the portal. If any of these ranks are available, the search is possible.

The format of a URL for GBIF is:

```
http://www.gbif.net/portal/ecat_search.jsp?countryKey=0&
nextTask=ecat_search.jsp?countryKey=0&nextTask=ecat_search.jsp&search
=terms
```

where *terms* is a taxonomic name (genus species subspecies). The following resource XML could be used:

```
<resource>
  <location>
    <if column="IdeGenus" filled="yes">
      <text>http://www.gbif.net/portal/ecat_search.jsp?countryKey=0&nextTask=ecat_search.jsp&search=</text>
      <value column="IdeGenus" format="url" />
      <if column="IdeSpecies" filled="yes">
        <text>+</text>
        <value column="IdeSpecies" format="url"/>
      </if>
      <if column="IdeSubspecies" filled="yes">
        <text>+</text>
        <value column="IdeSubspecies" format="url"/>
      </if>
    </if>
  </location>
  <location>
    <if column="IdeSpecies" filled="yes">
      <text>http://www.gbif.net/portal/ecat_search.jsp?countryKey=0&nextTask=ecat_search.jsp&search=</text>
      <value column="IdeSpecies" format="url" />
      <if column="IdeSubspecies" filled="yes">
        <text>+</text>
        <value column="IdeSubspecies" format="url"/>
      </if>
    </if>
  </location>
  <location>
    <if column="IdeSubspecies" filled="yes">
      <text>http://www.gbif.net/portal/ecat_search.jsp?countryKey=0&nextTask=ecat_search.jsp&search=</text>
      <value column="IdeSubspecies" format="url" />
    </if>
  </location>
</resource>
```

The complete registry entry would be:

```
Group|Default|Table|ecatalogue|Resource|Search GBIF|<resource>
<location> <if column="IdeGenus" filled="yes">
<text>http://www.gbif.net/portal/ecat_search.jsp?countryKey=0&nextTas
k=ecat_search.jsp&search=</text><value column="IdeGenus" format="url"
/> <if column="IdeSpecies" filled="yes"> <text>+</text> <value
column="IdeSpecies" format="url"/> </if> <if column="IdeSubspecies"
filled="yes"> <text>+</text> <value column="IdeSubpecies"
format="url"/> </if> </if> </location> <location> <if
column="IdeSpecies" filled="yes">
<text>http://www.gbif.net/portal/ecat_search.jsp?countryKey=0&nextTas
k=ecat_search.jsp&search=</text> <value column="IdeSpecies"
format="url" /> <if column="IdeSubspecies" filled="yes">
<text>+</text> <value column="IdeSubpecies" format="url"/></if> </if>
</location> <location> <if column="IdeSubspecies" filled="yes">
<text>http://www.gbif.net/portal/ecat_search.jsp?countryKey=0&nextTas
k=ecat_search.jsp&search=</text> <value column="IdeSubspecies"
format="url" /> </if> </location> </resource>
```