Solr Date Querying and Ordering to Allow Date Range Searches

Introduction

The recent Open Access specification asks, among other search enhancements, for date range searches that will find all objects with a date between user-selectable start and end dates, for instance 1995 through 1999, or 2008 through the present. These dates will be derived, ultimately, from various dates described within the mods:originInfo element, such as mods:dateIssued, mods:dateCreated, mods:copyrightDate, and mods:dateOther. Many of these dates allow values such as "Spring 2014", "2014-05-15", "2014-05-15T01:30:56.044Z", "May 15, 2014", "and "Circa 2014". In order to do reasonable date ordering, all of these dates will be normalized in "sort fields" to a year-month-day date, e.g. "20140515".[1] These special sort dates, which are created in the Solr index alongside the text values in descriptive metadata fields, can be used in the various portals that require date ordering.

Proposal

The most common and elegant way to do date range searches is to use Solr's [startdate+TO+enddate] range operator. To accommodate such searches, we will need an improved set of date sorting fields based on the different MODS date elements.

Approaches

In the following, two approaches are taken. First, special sort dates are created for each of the MODS date fields that may or may not be found in a Fedora object. Secondly, a "super sort date" is created based on the algorithm suggested by the Open Access specification (page 2, section 4):

The date should match on <mods:originInfo> <mods:dateIssued>
Lacking issued date, go to <mods:originInfo> <mods:dateCreated>
Lacking create date, go to <mods:originInfo> <mods:copyrightDate>
Lacking copyright date, go to <mods:originInfo> <mods:dateOther>

It should be noted that while all the mods date fields mentioned in the Open Access specification are multivalued fields by definition, Solr can only sort on univalued fields such as the Solr ID or sortdate_i. Thus all sortable date fields are of necessity abstractions of the various dates possible in a Fedora Object.

A. Sort fields based on MODS Dates

The sort fields of the various individual mods date types (e.g., mods:dateCreated or mods:dateIssued) will be "intergerized" versions of the text values found in the latest instances of these elements in the Fedora object. That is, if more than one of any of these date elements occurs, the test falls through until the last element is chosen for the individual sort fields, dateisort_i, datecsort_i, datecdsort_i, and dateosort_i.

B. The Super sort Date

The super sort date, sortdate_i, is created by testing the object's dates in the order specified: 1) try to find a match for mods:dateIssued; 2) failing that, try mods:dateCreated[2]; 3) failing that try mods:copyrightDate; and 4) finally, try mods:dateOther. If more than one of any of these date elements occurs, the test falls through until the last element is chosen for sortdate_i.[3]

The Solr index user, therefore, will have various data sorting options available depending on the needs of a particular portal or collection. A collection such as Faculty Deposit, where all the dates are stored as

---

[1] Unspecific dates such as "circa 2014" are normalized as "20140101"; dates like "sping 2014" are normalized as "20140301".

[2] In practice so far, the vast majority of dates occur in the mods:dateCreated element.

[3] The B.C.E. dateOther dates in Roman Coins are treated specially and converted to negative numbers for the sortdate_i field to allow reasonable sorting. Thus 500 B.C.E. (-500) will sort before 240 B.C.E. (-240) in an ascending sort.

mods:dateCreated, might want to use the datecsort_i field based on mods:dateCreated. It is also possible to fall through different sorts in a search query using cascading Solr sort parameters, e.g.:

sort=dateisort_i,datecsort_i,datecdsort_i,dateisort_i

A more general portal or collection, however, such as NJDH or the Solr search of dlr/EDIT, where dates of one sort and/or another are likely to occur, or the Roman Coins portal, which has special needs met by sortdate_i, might prefer to use the super sort field, which will always have a date that can be used for sorting. All of these options are always available.

Issues with date sorting

The free texts fields of the mods date fields have allowed a wide variety of date formats to be input over the years. Thus there will be some objects where no harvestable date for sorting can be found. If no suitable date can be found, as in a collection object with no date or an object where a date is given as "unknown", "unkn", or some other variant, a sorting date of "99990000" is assigned assuring that the object, when sorted by date asc, will appear after the range of objects with valid dates. While nothing can be done with empty or "unknown" dates, attempts are made to clean other problematic dates, such as "ca. 1915" or "Spring 2014", including parsing such strings with the PHP strtotime function.

1) Objects with no usable dates get a sort date of "99990000".
2) Dates with the string unknown or unkn are converted to "99990000".
3) Circa dates are converted to the first day of the year: "circa 1915" is converted to "19150101".
4) Seasonal dates are converted to the month when the season begins: "Spring 2014" becomes "20140301".

In addition to sortdate_i, we propose creating the following sortable fields for all the possible date types: dateisort_i for dateIssued, datecsort_i for dateCreated, datecdsort_i for copyrightDate, and dateosort_i for dateOther. This will allow, say, a dissertation with a dataCreated of "2013" and a copyrightDate of "2012" to be sorted on either of these fields as well as the main sortdate_i values of "2013".

Examples

An object with these mods elements:

```
<mods:originInfo>
<mods:dateCreated point="" qualifier="exact">2008</mods:dateCreated>
<mods:dateOther type="degree" qualifier="exact">2008-10</mods:dateOther>
</mods:originInfo>
```

will get the following date fields in the Solr index:

```
<field name="dateother">2008-10</field> # tokenized text field
<field name="datecreated">2008</field> # tokenized text field
<field name="sortdate_i">20080101</field> # dynamic integer field
<field name="datecsort_i">20080101</field> # dynamic integer field
<field name="dateosort_i">20081001</field> # dynamic integer field
```

An object with these mods elements:

```
<mods:originInfo>
<mods:dateCreated qualifier="exact">2013</mods:dateCreated>
<mods:dateOther type="degree" qualifier="exact">2013-05</mods:dateOther>
<mods:copyrightDate qualifier="exact">2012</mods:copyrightDate>
</mods:originInfo>
```

will get the following date fields in the Solr index:

```
<field name="dateother">2013-01</field> # tokenized text field
```

<field name="datecreated">2013</field> # tokenized text field
<field name="copyrightdate">2012</field> # tokenized text field
<field name="sortdate_i">20130101</field> # dynamic integer field
<field name="datecsort_i">20130101</field> # dynamic integer field
<field name="datecdsort_i">20120101</field> # dynamic integer field
<field name="dateosort_i">20130501</field> # dynamic integer field