Making the most of your MARC record in Evergreen

June 9, 2020

>> DEBBIE LUCHENBILL: Hello, it is just after 2 PM, so welcome to track two on the Evergreen virtual conference. I would like to introduce Galen Charlton will give the first presentation making the most of your MARC records. I will monitor the chat and a Q&A and hand raising so if you have anything you would like to convey to Galen do that and I will keep track of that and let him know when he's ready to stop her questions. Galen?

>> GALEN CHARLTON: Hello, I am very happy to be here. Like Debbie said, I will be talking about how to make the most of your MARC records in Evergreen. What I will be doing with this presentation is mostly just scratching the surface. But hopefully in a productive way.

First, we will start by talking about the true purpose of modern iOSs and Discovery system. Of course, the purpose of the modern library catalog is really all about the catalogers. Modern library catalogs like Evergreen that make full use of indicators and fixed fields are an excellent correction tool for MARC records. It can be really obvious if your leaders -- leads to the wrong icon being displayed in the public catalog. Another way of looking at it is the modern library catalog is arguably the revenge of the indicators for having been, if not ignored, perhaps underfunded or not having enough attention paid to them over the years.

A more positive way of putting this is that the modern library catalog is a way to make the full use of literally decades of detailed work by many, many catalogers over the years.

Evergreen of course already does a lot with MARC records. Evergreen's indexing is based on the MARC records. The MARC records display -- drive the display of all aspects the public catalog and interface display and many attributes are driven, or extracted from the MARC records.

But what it will be doing with this presentation is showing you some ways that you can go beyond the box. By that, I mean the box of what is available in Evergreen, using the data. So what it will be doing is doing a deep dive into the record attributes system. Showing how it can be used for searching, how you can add custom record attributes. And how you can maintain them. One I will also talk a bit about URIs, a different topic but when that of course matters quite a lot for electronic resource management in Evergreen.

This would be a pretty poor presentation about MARC records if I did not actually show you one. So, I made the discovery recently that there is a totally awesome thing called Hades town that I would love you to see when they. Once circumstances are permitting.

What I have done is I will be using the record for the original cast recording of Hades town is the primary example for the rest of this presentation. It is nice about-- what is nice about this record is that as you can see it is one that has been kept up-to-date with evolution of MARC 21 and the application of RDA to MARC. That's a long-winded way of keeping your eyes on those 3 and 4X fields since we will be doing a lot with them. When I talk about MARC record attributes in Evergreen, what I am talking about is a system in Evergreen for inspecting the MARC record, and drawing sets of values that are for the most part controlled values things like fixed fields, particularly with the advent of RDA records so from the "of new fields" including various D33X's, 34X's, etc.

Representing the audience, language, the -- the language code, media_type from 33X, etc. these values I will explain here are what come with the record attributes that are stocked in Evergreen.

So, if, we then move onto the question of what is the point of it? The record attributes -- among other things -- let you do searches and filtering. So what I have here is a screenshot of a search of media_type parentheses audio. That makes use of evergreens search syntax for filtering the results, but since the search is in the filter it says, please show me all records, the media type is audio-.

You notice an icon for music recording and that is automatically driven by one of the stock attributes come in this case icon format, which is driven by fixed fields and is used by the public catalog to display and to select which material icon to display.

That we see what is possible with record attributes, let's go ahead and trade a new one.

So, one of the things that RDA has given us, of course, is a large number of additional-- MARC tags. So this I will focus on the 340 in particular, the 340 A the material base and configuration. Which is a slightly verbose way of saying what is the item physically made of?

So, this is a CD, a compact disc is made of plastic with some metal film, and in this case 4-three-quarter inches in diameter. Out-of-the-box, Evergreen does not do anything with the 340. So let's go ahead and figure out how to change that.

So in the Evergreen staff interface, if you go to server administration, there is a page for configuring record attributes and the record editor calls it the S VF record attribution single value although that is a slightly outdated -- terminology E. That they can be single value and they can be multivalue. If we jump back we see this particular record has two 340A's, both metal and plastic, Evergreen can be configured to store both of those as being values of the material base attribute for this record.

If we go through what the definition is, this is pretty straightforward. This is not composite attribute, a composite attribute is a way of saying you want to combine different attributes together into a meta-attribute. This is only pulling from a -- MARC field. The description is hopefully a helpful one. This is mostly for staff use. This is going to be acting as a filter, so, checking the filter box tells the Evergreen search engine that you do want to be able to limit search results by this attribute.

This is not a fixed field and we are not joining multiple instances. We can give it a label, the label is essentially whatever you want to be. And -- the instructions for where to drop this attribute is using a MARC tag and attribute. We checked the multivalue box because unlike, say, something driven by the leader, there can be more than one value for this attribute in a given record. And we have given us a name. The name here is one used in areas like the search engine.

Material_base has the name. So if we go ahead and move on, we can go to another aspect of configuration which is the coded value maps. The coded value maps were way of establishing a fixed vocabulary. And it is tied to an attribute. For the material base attribute, we have a plastic, very plastic, you can choose to have the value be a different display level. Display name, and similarly, one for metal,

So what could value maps to is a couple of things. One, is they serve as a way of imposing some constraints on your attributes. As in, it makes it easy under the hood to run a report to identify records that have an attribute but use of the value is not on the coded value or, dare I say, authorized list.

So, you know, it also serves a function for search performance. Values that -- for an attribute that are on the coded value map, are internally represented as numbers or integers in the database and the underlying search system, mechanism, that applies search filters based on attributes essentially can use the editors to very efficiently identify the records that meet a filter condition based on an attribute.

So, having said that, the coded value maps are useful for controlling vocabulary, but it's not a straitjacket. In other words, it's possible to say an attribute can be controlled, whatever you find the fixed field you are using, or whenever you find the mark-- MARC field will get represented in the attribute, exactly as it is stored in the MARC record. So, with that, you get both control and flexibility. Now, if you wanted to, you could define attributes for any MARC filled that you choose to. So, if you really wanted, you could in fact define a record attribute for the 245 A. However, that is not something I would recommend because it will not work particularly effectively as a filter. Nor will it be particularly useful in display purposes paired Evergreen I should point out, also has a concept of display entries. And that is something you can configure to say, okay, I want to wrap my title from a certain field and subfield, or I want to grab the title as expressed by the MARC record transformed into mods. And make it display where I need it to. And that, by the way is a mechanism that will end the attorney of t titles in the reports. Long-term Evergreen users I am sure have repeatedly run into situations where doing reports against certain record sources and displaying fields like title and author have the value shop strictly in lowercase. As opposed to what you would expect from values as they are stored in the MARC record.

Display entries, they fix that problem since they let you control how the display values are extracted from the MARC record, and they do not force lowercase on you.

Enough with that digression. Before I continue with the presentation, I want to pause briefly and see if there are any questions. For some reason I was able to see there were some issues showing up but not seeing the content of them. Debbie?

>> DEBBIE LUCHENBILL: There's a question about whether the slides would be available and I said, yes. And at the moment, there are no other questions, so folks, if you have any, please type them in the chat or the Q&A or raise her hand.

>> GALEN CHARLTON: Thank you, Debbie. I will say that if you do have questions, I suggest using the Q&A because unlike the chat, for some reason, I am able to see the Q&A. -- Let's go back and move on. I say that, the chat window bubbles up to the top for me. Okay. Great.

Technology is wonderful. Isn't it, folks? [Laughter]. Now that we have coded value maps, in place. I was talking about controlled vocabulary. Well, that's actually something we can do in the MARC editor. Evergreen does store tag tables in the database, they define the tags and subfields and the display labels for them.

But one thing they can also do is associate a coded value map with a mark subfield.

I did this for my example, and as you can see from the screenshot, for fields like the 340, you can make it possible to right-click and get a list of values for the subfield. This by the way, something is not tied specifically to record attributes. This is something you can define for any field in the MARC record.

For example, if you say, you want to add notes to records, identifying the source of a donation, you could define a drop-down for this to say, okay, rather than having to type out each time with a full donated by the society for the relief of lost ducklings, you could associate a value map without subfield and select it from a drop-down.

This is not exclusive, even if a field has a value map associated with it, you can also type in every you want. And so, the theme of control and flexibility applies to this mechanism.

You will notice a big asterisk in the slide title, as much as I would like to blame my cat for that, the asterisk represent a caveat, and that is at the moment, the way you would associate a coded value map with a MARC subfield, is by doing a database update. In this case, specifically to the table config.MARC subfield. There are interfaces and server administration to change the tag tables, but as I discovered while I was preparing this presentation, they are a little buggy and will require a little work.

That said, if your local friendly Evergreen administrator has SQL axis, it's an easy update to associate valueless tags and subfields.

Moving from the staff aside MARC editor to the public catalog, what I am showing here is the result of defining the attribute. That is that you can do a search, the name of the attribute in this case the value you are looking for in parentheses which gives you away to identify the items made of metal using material_base(metal).

This is something was done without requiring any code changes whatsoever. So, if there has been some new MARC tags you have been looking at and wondering if you'll ever be able to make use of them for Evergreen, the power and capability already exists to do a lot without having to write a single line of Perl or JavaScript.

That is pretty useful. And of course, one of the things on a policy level that this can lead to is the ability to say, if you are doing a MARC cleanup project, were specifically -- or specifically a record enhancement project to say, not only are we bringing in records that adhere to modern cataloging standards but were able to drive specific user functionality with them. We have a question, I would like to create a record attribute for subfield three for audio enabled and 14 347 subfield be for the MP3 audiobook. Can I create a filter that looks only at that particular tag and subfield?

The answer is yes. Depending on exactly how you want to do it, you can create two attributes, one drink from the 347 subfield three, and another drawing from the 347 subfield B. When to create that filter, attribute rather and once you have created coded value maps for those attributes, although that is something that is optional. Either way, those attributes, almost immediately, available to create search filters. The way you are accessing that filter is by typing in Evergreen search syntax. But that's not the only way to do it. You can also, with some customization of your OPAC templates choose to make an OPAC drop down for material base. Or you could do things like establish permit links to say that I want to have a search such as all the glass globes my collection.

Now, in the example of the question about the, two subfields from the 347, what you potentially could also choose to do is to consider setting up composite attribute that draws from both subfields.

Thank you for the question. So, we have discussed searching. But before I continue, I will now talk about bids that I have hand waived, that you would see I hand waived if I had the video camera on and there is a bit of work that you have to do after you define an attribute.

So, if you create an attribute-- Is something I would like to take from my MARC records. But defining the attribute is not automatically create the actual attributes for each record. You do have to index it. So one way you can do that as you experiment with an attribute, simply saving some records that have the field you are interested in.

That is away to make sure the attribute is behaving as you expect it to. And once you have finished testing, the next step you have to do is to index it. Across all records.

In this case, I am jumping -- one way of doing it that is accessible to, again, your friendly local Evergreen administrator, of doing SQL, there's a stored function called meta-bibattributes. please readjust the indexed, all attributes on a record or in the case of the top example, just -- in this case using material base, the specific name you're interested in.

The entire database, you can run this function for every record that in this case has that has the field and subfield we are drawing from. And there are other ways of doing this, you can choose to do a complete reindexing on all the fields of the entire database, but that tends to be overkill just for defining a new bath-- attribute.

When you define a new attribute in order to use it as a filter sometimes you may need to restart Evergreen services in order to have the search engine recognize it. Okay you've defined a new filter. So moving on -- out of handwaving and searching onto display, one of the things you might want to do with a new attribute is make it show up in the catalog. This would be an example of displaying 340 A. I should've mentioned this presentation is aimed at cataloging staff and the Evergreen SysAdmin staff and for folks that were both hats. In this case for a -- for looking at the OPEC template on the server -- let's display the material basis and give this a label called material bases and let's display the values, material_base.join is the display of the attribute values, separated -- by commas.

You might ask, material bases it sound like it's going for the record attributes I have been talking about. And it sure does. Sadly, no, it is coming from a different mechanism. The bit of code extracting the 340 subfield A is actually looking at the entire -- MARC record using the XML fine nodes to extract all 340 subfield and sticking them in a list we can display later. This is an example of cases where there's more than one way to do things in Evergreen, so record attributes you define and server administration work in search and filtering but the display is more driven by what the opaca templates are extracting out of the complete -- MARC record. Two different mechanisms and it means is if you are using a given attribute both for display search, that you will have to deal with two mechanisms and keeping them in sync. That said, not ultimately too troublesome in the default case of picking up on one of the new RDA subfields and making use of it.

For complete change of pace, next we move on to talking about located URIs. This is something that is particularly important for consortial users of Evergreen that use electronic records into a catalog.

Most of the time would be grand to say that your electronic resource subscriptions and their pricing meant the consortium created-- buy everything for their libraries. In practice you could, that is not always possible or feasible and of course, different libraries have different communities that they need to serve. And so, what the located URI mechanism does say you have a URL for an electronic resource, that is stored in an 856 field, URL and subfield, and you can add as subfield nine or a lot more subfield nine's that have the short name of the organizational units that electronic resource belongs to or should be displayed for.

Those control what shows up in the electronic resources box in the public catalog. The combination of what subfield nine specifies as well as what you have configured for the search context for each library. cases where fixed fields have not necessarily been fully used by software.

So in one sense, the definition of a fixed field has expanded. Almost all the new RDA tags have implicitly or explicitly controlled values associated with it. If you look at fixed fields as the concept more of not necessarily fixed position, but a field that has a relatively small control field vocabulary associated with it, you have a lot more to play with. Cleanup and enhancement projects matter because fixed field matters. Fixed fields matter. But it also is something where you can turn a cleanup project, not just into something that gives you consistently, consistency, but drives additional functionality in your catalog.

Also, if you want to filter something, you can do it. You can do it with no coding required and relatively little effort on the part of your friendly local Evergreen SysAdmin.

So to recap, record attributes can be used out of the box, but you can also customize them and use them to drive searching and filtering. XPath, which is what the OPAC template and its fine nodes invocation was using drives the display for the sorts of attributes. And located URIs drive electronic resource visibility.

Like I mentioned at the beginning, this is just scratching the surface of what you can do with MARC records and Evergreen beyond just what is in the box.

So, thank you for your time and attention. I think we have a couple minutes left for any questions.

>> DEBBIE LUCHENBILL: I am not seeing any questions or hand raising anywhere. Or in the chat.

>> GALEN CHARLTON: Okay, thank you again for your time and attention. And yes, the cat you just saw is indeed the George -- is indeed George. Apparently there is a memo going out that every presenter during this web conference would be issued a new kitten, so here you go. You can all right, somebody is asking a question would like directions with screenshots for creating the record attributes.

So, the presentation will be recorded and available on YouTube and the slides will be made available on the Evergreen website, but I can also expand a bit elsewhere on going through all the steps required to define a new record attribute.

All right, another question, when is the cat photo exchange portion of the conference? There is indeed a social event in the conference so there is your opportunity. All right, with that, again, thank you for your attention. And we can take 12 minutes before moving on to the next session. In this track.

>> DEBBIE LUCHENBILL: Thank you Galen and I will be getting a slide up for the next presentation, shortly. So stick around, we will just keep this track running. So if you are not planning to attend the next session, please do close out of your Zoom so there's enough seats for people who want to attend that session. I hope to see you at the next one.