CAS WEBINARS

EQUINOX TRACK 2: IF THIS THEN THAT: ACTION TRIGGERS ARE MORE THAN JUST NOTICES

JUNE 11, 2020

CAPTIONING PROVIDED BY:

CAPTIONACCESS

contact@captionaccess.com

www.captionaccess.com

>> Good afternoon, everyone. We are going to get started with this program. This track is sponsored by NC Cardinal with captioning made possible by Equinox Open Library Initiative. And we would like to thank our captioner.
We would also like to thank the other conference sponsored for making this event possible. Mobius, Bibliomation and Evergreen Indiana. The event is being recorded and will be available on YouTube following the conclusion of the conference. We would like to encourage ever want to use the chat window to post questions. The facilitators will be collecting questions along the way and causing them to the presenter at the end of the session when she asks for questions.
We would like to introduce Michele Morgan from NOBLE, the North of Boston library exchange, who will be present If This Then That: Action Triggers are More than Just Notices.
>> Okay, thanks, April. My name is Michele Morgan, and I'm the technical support analyst at NOBLE, the North of Boston library exchange. Just to give you a little context, here is a quick slide about our NOBLE libraries. We have 17 public libraries, 8 academic and one special library, and we are situated north of Boston in Massachusetts. So when we talk about action triggers, we need to maintain those for our libraries. So one question is, what action triggers do? They do a lot.
They send a letter email notices, overdue, messages related to holds, welcome messages for new patrons, messages for soon to expire patron records to remind them to update their accounts.
Password reset e-mails use action triggers as well as when you email yourself a bibliographic record from the catalog, that's an Action Trigger. The checkout receipts in the web client, those are generated by an Action Trigger. If you have them enable, their SMS or text action triggers that can send hold ready for pickup messages. There is a stock trigger for an SMS courtesy notice when your items are due in the next few days. And you can also text yourself conifers from the OPAC.
Acquisitions makes heavy use of triggers, which I'm not going to talk about because I don't understand them. That's a presentation in itself.
Also, action triggers do things like process auto renewals if your system uses that. MARC items lost or long-overdue, and that involves building a patent for an item that have had out for too long. If you use the Evergreen self check, the built in one, the receipt generated by the self check our action triggers.
Lots of the print functions in the web client in the OPAC have an Action Trigger underneath them. You can use action triggers to put a message for

a patron in their account when they log into the catalog in the message center. Then delay or mark import export front and use a lot of triggers. And all the emergency closing processing is done by triggers.
Soy want to start by looking at all the bits and pieces that go into triggers. So if you have permission, and that's a big question, action triggers send a lot of email. So you really want to think about who has permission to get into them and change them. But if you do have permission, and you go to administration, local administration, and choose notifications/action triggers, you get this screen, which has all of your event definitions listed here. And they should look familiar. This is a stock system. But you can see it has courtesy notice and overdue notices and auto renewals here. So the tabs across the top are the event definitions, the next tab are the hooks, and that's the if this part. What the hooks do is they -- some of them listen to what Evergreen is doing and if an event happens in Evergreen, like some of the checks out and item, that is a hook. And you could use that to make Evergreen do something, based on that checkup. He took has what is called a core type, and that is a data source in Evergreen. It's the name of a class in Fieldmapper for that XML file that we will look at later.
There is a description there which is pretty good, to give you an idea of what they do.
This column, passive, most of these are not passive triggers. There is only one here that is a passive trigger. The difference is the triggers that aren't passive, and action has to be done in order for them to happen. Like you click on a print button in the catalog to print a bib record. And that's an active trigger. A passive trigger is when you have a checkout item that is due in three days, the system recognizes those as time passes. So you can use the checkout trigger the checkout to trigger and send a courtesy notice, and when that becomes true, that's when that trigger gets activated.
The next tab is the reactors. That's the gang that part. So based on those hooks, you can make something happen. A lot of the things you make happen are you send e-mails. But there are a lot of other things that can be the reactor for your triggers.
And the next tab is the validators. This is important because you could have situations where something was true when you're trigger was generated, but by the time it's time to process it, is not true anymore. So we always want to check and make sure the initial condition is still true before we send that email for something that somebody already returned.

So if we go back to the event definitions, each name of the definition is a link so if I click on, say this three day courtesy notice that is at the top, I get another set of tabs. The first one is the environment. And the environment, it makes data available to your template. So if you are sending an email message, your taking pieces of data from Evergreen, and you need -- need to make sure that your Action Trigger knows where that data is coming from. It uses paths to find the fm.idl.xm, also known as Fieldmapper. And getting environment right is important. Because if you leave something out and you call for this data in your template., it's going to be blank. Nothing there.

Also if you put it in there wrong, it's going to cause your trigger to fail, and nothing will run.

So here is an example of an environment that we use in our two week overdue notice and any of our notice -- notices with items. One goal here is to print every piece of the call number with its prefix and suffix and parts, if they are in use.

So how do you fill in these paths? I have a few brief slides here. If there's time at the end, we can look in more detail at the Fieldmapper, but I just cut and pasted pieces of the Fieldmapper hopefully to give you an idea of how to get those pieces of data.

So in the checkout do hook, it's looking at the data source -- as circ, which is circulation. And one of the things that he circulation links to is the private copy, the item that is checked out. And that shows as a link. And further down in the definition, that link is defined. So target copy, links to another data source called ACP. And if we find that data source, ACP, which is the copy, here is the database table that points to, one of the fields in their call number, which is also a link.

So if I follow that down further in the definition, that takes us to the class ACN. So when we go there, we see here is the label, which is what we are looking for. And that is a text. That is not another link. So we are as far and the environment as we need to go pick we can just grab that text.

So if we add to our environment target copy.call number to the environment, we can call in our template, circ target copy call number little look label.

Okay. The other tab here is parameters. Now, many of the triggers not have parameters. This particular one is a mark long-overdue trigger. And that is things like changes the status of an item to long-overdue. So you need an editor variable to be able to do that. So that is why that is there. This is a good tab to know about because it lets you do a test of a trigger if you are working on it. But having barcode of a checkout copy. So you

could check an item out to yourself and plug that barcode in here, and if it's an email trigger, when you press go, you will get an email, a test email, that shows you what your output is going to look like. It will also show on the screen. But it looks a little weird. The formatting is weird. But you will get the email if that's the way your trigger is set up.
So back to the event definitions. If you double-click on one of these rose, like the first one, you get a whole list of fields that you can configure for your event definition.
This is a three day courtesy notice. We already talked about the hook. It's enabled. Processing delay and Max event validity delay work together. Also with the due date. So what this is saying is for this event definition, we want to look at items that are checked out and do between two days from now and three days from now. And we are going to act on those items and send the person an email to let them know that there items are due soon. Our validator is the item is still checked out. The definition ID, that's how the database keeps track of where this event lives. We will look at that a little more later.
I'm just wanting to go through the rest of the fields here because we will talk about them later. You can also message using a trigger. Those are the fields for that. You can also do opt in triggers which we will talk about later, and set retention intervals also later.
But then we come to the template, which is the part that actually generates the message that goes to your patron.
So if you look at that a look closer, these are written in template toolkit. The little things between the square brackets with the parentheses are template toolkit commands and they can assign variables or call data. But interspersed, you can see there's just plain text. So there is a lot that you can do here without knowing much template toolkit. For example, if we look at this, the salutation, it says dear user family name, is her first name. So dear Jones, Bob. Maybe I want that to say dear Bob Jones. So I just swap those around -- okay, I missed something. Maybe that’s later. Sorry. Let's move on.
This slide talks about some helpers that you can use in your templates. There is a way that you can format a date in a number of different ways. There is a helper that will get the copy price and what it knows how to do is go to the copy, find the price. If it's there. If it is 0, it will look at the library setting that will I allow you to have a default price and it will use that instead. So it knows how to do that.
There is also a helper that can get basic information about the items that you're going to print in your templates, like title information, author intimate

information. You can get org unit settings with a helper. You can get user settings from a patron record like the default SMS or text number that the patient has as a junior setting. You can also generate the SMS Gateway email with a helper.
So we are going to look a little deeper under the hood. First we are going to look at some database tables. And these are the database tables that relate to action triggers. And a lot of them you will recognize. The event definition is what we were looking at in the client. As a matter of fact, most of these we already looked at through the client's eyes, which is fine. But there are a couple that we didn't see there. The first one is called event. So Action Trigger event has these fields. And this is really important one, because when we are doing the three-day courtesy notice, and we gathered about 50 items that need to be processed to send these messages, each one of those items is going to generate a role in the event table. And the Action Trigger process will work through those rows and process them.
The other table we can see in the client is the event output table. And this data field is where the generated email actually lives in the database. So you can access that if you need to. And you also need to be aware that that could be sensitive data. So it's safely tucked in the database, but it is worth noting that that data needs to be managed.
Just briefly, here is the event definition table. I'm not going to talk about this because it knows exactly what we were looking at in the client. I do want to apologize for this next slide, but I wanted to try to give you an idea of how these tables all linked together.
Here is the event table. This is the one that contains those 30 items that we will preview notice. And it has a state and as the triggers run, this road changes and reflects different states of the trigger. Points back to the event by its definition ID, which in this case is 38 won. The .380 won. The event likes to hook, reactor, and validator. Like we saw before. And also the environment by its ID. And not a state event table has a template output field and that has an ID for the event output table. So that links those together.
One thing I want to call your attention to before we leave this slide is the event ID, which is something like 35,500,000 et cetera.
So more under the hood are the Perl modules. When you have the reactors and validators, those all point back to the Perl modules.
See it? Another piece is the crontab, which tells Perl when do you want to run these. This is pretty much a stock won. The example that Evergreen

comes with, and by default it will run all the triggers every half hour. Which is fine.
Further on down here are the stock granularities. If you want to run them at different our jewels, like hourly, daily, those all there.
And the third crontab entry here is the one that purges the old events. Which we will talk about more a little later.
One more piece. You can also use custom filters to limit -- to limit the events that are generated. For example, you might not want to create an email Action Trigger if the patient doesn't have anemia address. Because it's going to sit in your database and it's going to end up invalid. So I have it in the 1st place? You can also use filters to send notices only to certain categories of patrons. Based on the data into patron record. And it's nice to use filters to avoid creating events that will just end up in an invalid -- invalid status at the end.
Filters use JSON query syntax. I have a couple of examples of filters that we use in NOBLE. The first one is the one that we use for processing auto renewals. We have libraries that opt into auto renewals and some that do not.
So the top is just a basic filter that Evergreen comes with. But down here, we are saying only run this for circulations that have in the circ lib, which is the place it was checked out, these unit numbers. And the other thing is if there is an auto the auto renewal raining field, we don't want to processes trigger at all. Because some circulars are just not renewable, and the trigger processes and it fails, by default it will send an email to a patient saying this is a nonrenewable item. I couldn't auto renew it. So we just autos to go.
Here is another filter that we use for our patron welcome messages. We have public libraries and academic libraries. For the academic libraries, mostly they are student records -- their student records are loaded from the bursar. So we don't want to send them a welcome message. Because they will get that information -- we don't want Evergreen to send them a welcome message. So we limit that by patient profile and also by the home library of the patron record.
Another filter. This is for an overdue notice that we created for reserve items. We had an academic library that wanted a notice to go out the next day for a reserve item. And like anything, when you are configuring things, there are a number of different ways that you can do it. But the way that was more straightforward was for this library whose or unit is 30, in the OpenSRF table, look for any items that have a duration of two hours,

which is the reserves. So it will send the overdue notices the next day for those items.
Okay for we can pause for questions. Let's see.
>> We have a lot of commentary from everybody who led the slide that you apologized for. Everybody thought that was fantastic.
>> I was thinking it might what be something you want to blow up and put on your wall. Benjamin had a question, so you have auto renewals turned on for everyone but use this to prevent some systems from having their items auto renewed? Regarding filters?
>> Yes, we use the filter in the trigger to -- I forgot about that part. You have to have the circ rules configured to do auto renewal and also the filter for the triggers to limit the ones that are processed because you don't want to try and process auto renewals for those that have not opted in.
>> Is not opposed to building individual triggers for each library that is using?
>> That's just like anything else, there are probably a million ways that you can do it. It just seems the way we instituted it, it just seemed to grow sort of organically from a single trigger that we filtered. You could do it either way.
>> Anybody else have questions you want to type into chat? Jennifer asks, where you put the filter code? In the template or somewhere else?
>> Good question. The filter code does not going the template. We have a folder in open ILS were all our filters live. And when you really trigger, the code that runs the trigger can call the filter at the pathway. So it was somewhere in the file system that you can call in the command.
Okay. I will move on we can take more questions at the end.
So I have a few trigger tweaks. This one got used a lot. Disable a trigger. When libraries started closing, we didn't want to send overdue notices, we didn't want to send preview notices, hold notifications. So by opening up the trigger, unchecking the enabled box and saving it, you trigger what run. So that's a good thing to know about.
And here's the slide that I started talking about earlier. Here's where I wanted change the text from saying dear last name, first name, to say dear first name, last name and add a sentence at the bottom that gives them a link they can log in and offer their account.
You can change the processing delay in a trigger. For this example, this is the cold ready for pickup notification. And by default, the processing delay is 30 minutes. And what that means is you check in an item, it goes on your hold shelf, that row in the event table is added, but it is not allowed to process until half an hour has gone by.

The idea is you need to make sure you get it on the shelf because another patient is going to walk in the door as soon as they receive that email message. If the item is in such a condition that you can't let it go for the hold, you need time to undo that hold so that by the time the cold notifications are processed, the fact that it is on the hold shelf is no longer true, so it will be marked invalid. So that's the idea of processing delay.
If maybe 30 minutes is not enough or too long, you can change that. Maybe you need an hour. So you change that and save it. And now behold notification event will sit in the event table for an hour before they are eligible to be processed by the crontab.
So let's look at password resets. So that they trigger, and the default processing delay for this trigger is one second. Which is great. We love those to go quick.
But by default action triggers only run every 30 minutes. So if you have -- your password reset and action triggers are going to run in a minute, you're good. You're going to get your email right away. But if action triggers have one a minute before you request your password reset, you're going to have to wait 29 minutes to get your email.
That is fixable. What we did is we just run all of our action triggers every 2 minutes. So we changed our crontab from 30 minutes to two comments. So our password resets will go out no letter then 2 minutes after they are requested.
The other thing you could do is give your password reset trigger a granularity. Can be free test, anything you want, something that you recognize, and you add a crontab entry that refers to that granularity, PwC__ reset is what I used and that will run only your password reset every 2 minutes.
So here's a message center message that we have added to the long-overdue Action Trigger. The idea was your charging a patron for a but that they have had for six weeks you sent them a notice, but it might be nice if they had a little more information, if they would log into their accountancy they all money, but that money is for. So we added that, that's part of the trigger. So the trigger sent out the notice and also adds the message that the patient can see. In the catalog when they log in.
We did a similar thing with the labor accounts card soon to expire, to try to let them know how they can renew their car.
So let me pause for questions again. Part of the command to run the trigger has to refer to -- and we will see something later.
Thanks, Jason, for printing out the default Action Trigger filters. Jessica says we just reduced the delay for hold notices for a lot of libraries

because a lot of libraries want to check the item out to the patron soon after the hold was triggered for curbside.
Reduce the delay. So you wanted them to get the message that there hold was ready and then the checkout would happen? That's interesting. That should work.
Are there any other questions? Before I go on?
>> I didn't see any more in the chat.
>> Okay. Let's go on. Let's talk about retention. In release 3.0, the ability to set a retention interval for -- and trigger events was introduced. Without those retention intervals, the event table just gets bigger and bigger. You remember that table would be 35,500,000 ID. That means that many events have been generated in the system. And if they are not getting purge, they are going to still be in their database, taking up room storing patient information that you don't want stored there.
So it's great to be able to purge old events from the database.
So to set that up, the first thing you have to do is decide how long you want to retain things. We decided on six coat months for circular should notices must because there are places in the client that display how many notices people have gotten. So if you sort of had a record if you have a long billing dispute that yes, we sent notices out, most triggers, though, we don't keep very long at all. If you print a record from the catalog, we don't want to keep that longer than one day. If you need to do a password reset, we don't keep that.
So once you have decided on your intervals, can add them to the event in the retention interval field, and you make sure that the entry that purges the event is enabled. And then your database will be clean.
So on to cloning triggers. You might need to create new triggers. You can create new triggers from scratch, but it's much easier -- easier to clone a trigger that already exists because chances are you want to make some changes and heavy trigger that is similar to one that you already have. So it's easier to clone.
Some cautions. Triggers love to send e-mails. So we want to avoid sending e-mails that we don't really need to send. So when you are working on triggers, a few ways to avoid doing that is if it is an email trigger, you can change the reactor to process template for testing. But then if you do want to see that output, you do need database access so that you can query it to see what that looks like.
You could, in your template, replace the email variable that gets the patron's email and hardcode it with your own email address so you will get all the e-mails.

If you're working on a trigger over a period of time, keep it disabled unless you are actively working on it. Just to avoid the possibility that it might run unintended. You can always add a unique granularity so the trigger will be called by the crontab. It can be a nonsense thing. Just say it's different from all the others.
And is important not to bite off more than Evergreen can chew. If you decide you want a trigger that needs a large it all between the processing day and the next validity, you can generate a whole bunch of events that your system can't handle. Yes, I have done that. But only once or twice. Ideally, use a test server to do your testing and then copy things from one server to another.
Okay peaks of the process of cloning triggers, you need to have a checkbox next to the trigger. You click clone, and I have always found that when you do it the first time not all the figures get filled in. So I do that, click cancel, and do it again. And then I get my fields filled in except granularity. I never get that filled in, but that's fine, I can fix that.
But we did, we used to have one hold notification trigger for the entire network. Some libraries want them turned on at different times than others at this point. So we have been cloning them. This is the system-wide trigger. I can change it to just be owned by one library. In the default text we can replace the default text with text that libraries are encouraging people to call so they can arrange a pickup appointment.
So when you say that, it will ask you if you want to clone the event definition environment. And why would you not?
When you are working with triggers, there are strategies that you can use to test and monitor. I mentioned the test by barcode. Those for work for the overdue triggers, the preview triggers, the ones that use circulation. So that's a handy thing to be able to do. And that also works if you're trigger is disabled. So you can keep you're trigger disabled and keep tweaking it and use this testing box to get your email.
So here is a way to run a trigger from the command line. This is a shell script. They copied a little bit from the crontab, and here's the actual trigger that runs. So here's the granularity that I have in the trigger. And here's the reference to the custom filter that I have for this trigger.
So if you are running a big trigger, I like to monitor it to see how those events are turning -- churning to be sure that I'm not biting off more than Evergreen can chew. So I used PG Admin 3and I am sad to hear that it's discontinued. But if you run that, you can see how the events are getting processed. So when he got separated, they start as pending and they go through states like collecting, collected, there are in number of states, but

ideally they all end up as complete. So when you trigger is done, this is what it looks like.
Blake says PG Admin 4 is alive and well. So this is a way you can monitor your triggers.
If you have changed your reactor to process template from send email, the output that you want to see is sitting in the database, this is a query that can get at that. It will select data from the output table, and it's probably a sloppy query. There is probably a better way to do it, but it works.
I'm going to go through the rest and then we can do questions at the end.
The next thing I want to mention are the opt in triggers. You can make triggers such that patrons have the option to opt in. In order to do that, you need a user setting type. We created three different user setting types because we wanted patrons to be able to opt into getting free due notice that was seven days before the due date, one that was five days before the due date, and one on the day that the item was due. So we created those.
And then you need to add those settings to your template. So here's the user setting, and you need this opt in the user field just because if you don't have it, I forget what happens, but it was something bad. So you need that.
So when you have done that, and you go to the patron just ration screen, you'll see choices that you can opt patrons into those triggers. patrons will also see those in the catalog. They can check them off and save them. So we have been doing that for quite a while. It seems to work out well.
A few MARC tips, tricks, and examples. We have libraries that regularly to food for fines during the holidays. So it was always kind of a pain, sure, I'm happy to add this to their trigger, but then I have to remember to take it out again. So by putting in a conditional, and using the date using the current date and comparing it to when you want the trigger to stop, you say if it's later -- sorry. If it's earlier than December 27, 2019, print this. Otherwise, print the regular text. So that's good.
And you can use conditionals if you have one trigger that different libraries want different text, you can use conditionals to give them that. So for org unit 58, they get this text. Org unit 67 gets this text. And everybody else gets this one.
Okay. Preferred names. When the preferred name develop it was done, the preferred names were made available to action triggers, but generally action triggers don't get updated because people customize them so much. So we have updated all our action triggers to always use the

preferred name or if there isn't one, use the first name, same with the last name.
It next and I notice we always, whenever we print the call number, we always include prefix, suffix, and parts. So this is this, number piece there is what locks that all in. And in order to make sure all the data is there, here's what you need to have in your environment to get that.
So those were my examples. I have some references that I just want to point out that they are here. And I also had added some wish list bugs, just to gather them all together. But I don't have to go through those slides, so if there are other questions that we can address, I'd like to do that now. So any questions?
>> To questions and if you have them. Meg likes the don't bite off more than Evergreen can chew reference. And you have probably done that. That's one time or another.
And there has been chat about PG Admin 3 versus PG Admin 4. So there are links and therefore the desktop app and the browser. Andrea posted another cheers for your table linking slide. Everybody like that.
>> Good. I do like the idea.
>> Blake said show it again.
>> All right. Let's go back there. And Benjamin says thank you. This is been very helpful.
>> Here it is.
Terran says they had to change a lot of theirs to run once a day to reduce server load. I can see that having the filter used might save buses running 20 triggers separately. Like says thanks. Good to hear your voice.
>> Thank you.
>> Tiffany says great job.
>> And Jeremiah said in the middle of retroactively creating trigger events around current events_dev has been turned off and resetting others to go again with changed template. Joy. We have a lot of that without various changes with COVID 19 as well.
>> It's been an opportunity to be creative with triggers.
>> Yes.
>> Jessica says super helpful. Jason says excellent presentation pierced.
>> Thank you. Before we go I just wanted to point out in terms of finding paths for the environment, I struggled with that for a long time. If you use these links, you can get a web representation of your Fieldmapper. And that you -- you still need to put on your glasses, but it will let you easily -- well, not easily. It will let you trace the paths to find your environment. Thank you.

>> Thank you very much, Michele. Good stuff. I will go ahead and grab this screen for the next session. Feel free, if you have any questions, if you have anything else you want to ask, please do. Next up we will have the SysAdmin interest group. If you hanging around for that, feel free to stay. If you going to another session, feel free to check out. Again, post your questions if you have them.