Learn Action Triggers in 30 Minutes or Less!

Chris Sharp
PINES System Administrator
Georgia Public Library Service
Prerequisites for Successfully Using Action Triggers

- A good idea about what you want to accomplish
- A tolerance for unfamiliar terminology/jargon and for source code(-ish looking things)
- Access to your Evergreen server and/or a decent working relationship with your system administrator
Why Action Triggers?

• What problems do they solve?

1) Evergreen needs certain things to automatically happen at certain times (e.g., notifications sent at a certain time every day).

2) Evergreen needs a way to automatically generate output when particular actions are taken (e.g., holds pull list requested, password reset via the OPAC)
What are Action Triggers?

- A system of user-defined events that are run automatically by the system at specified times and/or when certain actions are taken.

- Examples:
  - Hold/overdue email notifications
  - Password reset requests in the OPAC
  - SMS functionality
  - Hold requests pull list
  - CSV Output for various interfaces
Getting Around: The Interface

- Admin → Local Administration → Notifications / Action Triggers
  - Double click the row to view/edit the event definitions.
  - Click hyperlinks to view/edit the event environment.
Setup: Event Definitions

- What do you want Evergreen to do?
- Best to know this before you start
  - “when X happens, I want the system to do Y”
  - Consider event parameters:
    - What should trigger the action
    - What action should be taken
    - Wording for templates, etc.
Setup: Hooks

• What will “trigger” the event?
• Examples:
  – circulation comes due
  – item is checked in
  – Item is checked out
  – user requests a CSV file
Setup: Hooks

- What will “trigger” the event?
- Active vs. Passive
  - “Active” - event creation occurs in real time
    - e.g., Item scanned for a hold, patron downloads CSV version of a list
    - happens within Evergreen itself
  - “Passive” - event occurs without any action causing it beyond the passage of time
    - e.g., Item becomes due, hold request expires
    - timing of event creation is managed outside of Evergreen itself (e.g. scripts that run on a utility server)
Setup: Reactors

- What action is triggered?
- Examples:
  - SendEmail
  - ApplyCircFee
  - MarkItemLost
  - ProcessTemplate
Setup: Validators

- Is this still something we want Evergreen to do?
- Examples:
  - CircIsOpen
  - CircIsOverdue
  - HoldIsAvailable
  - HoldIsCancelled
Trigger Event Definitions

Max Event Validity Delay
11 days

Opt-In Setting Type

Opt-In User Field

Success Cleanup

[% USE date -%]
[% user = target.0.usr -%]
To: [% params.recipient_email | user.email %]
From: [% params.sender_email | default_sender %]
Subject: Overdue Notification

Dear [% user.family_name %], [% user.first_given_name %]
Our records indicate the following items are 10 days or more overdue.

[% FOR circ IN target %]
  Title: [% circ.target_copy.call_number.record.simple_record.title %]
  Barcode: [% circ.target_copy.barcode %]
  Due: [% date.format(helpers.format_date(circ.due_date), '%Y-%m-%d') %]
  Item Cost: [% helpers.get_copy_price(circ.target_copy) %]
  Total Owed For Transaction: [% circ.billable_transaction.summary.total_owed %]
  Library: [% circ.circ_lib.name %]
%n %n
Field Definitions

- **Owning Library** – the library at/for which the event is processed
- **Name** – a unique name (per owning library) for the event definition
- **Hook** – the attribute upon which the event “hangs”
- **Enabled** – turns the event definition “on” when checked
- **Processing Delay** – how long to wait before processing the event
Field Definitions

• **Processing Delay Context Field** – when applicable, the date/time field used to calculate the Processing Delay (e.g. due date)

• **Processing Group Context Field** – defines how the events are grouped – e.g., prevents 7 emails for seven overdue items for a single patron. “usr” (abbreviation for “user”) is the most common

• **Reactor** – the “action” that is “triggered”

• **Validator** – means to verify that the event is still valid (e.g., the circ is still overdue)
Field Definitions

- **Definition ID** – the internal database ID for the event
- **Granularity** – optional means for “tagging” events so they may be processed separately
- **Max Event Validity Delay** – the “outer limit” past which we no longer want to consider matching events valid
- **Opt-In Setting Type** – for “opt-in” features, such as patron hold/circ history
Field Definitions

• **Opt-In User Field** – for “opt-in” features, the field used when determining whether a user has opted in

• **Template** – The template-toolkit-defined template for this event.
Triggered Events

- From a patron account, go to Other → Triggered Events
- Provides a method for front end staff to check whether/when a notice was sent, for example
Additional Resources

- **Evergreen 1.6 Documentation:**
  http://docs.evergreen-ils.org/1.6/draft/html/actiontriggers.html

- **Evergreen Dokuwiki:**

- **Evergreen mailing lists and IRC channel**