

? Question

What are the possible ways to trigger the notifications



by jaiku

created 2y ago (edited 2y ago) in Service Management

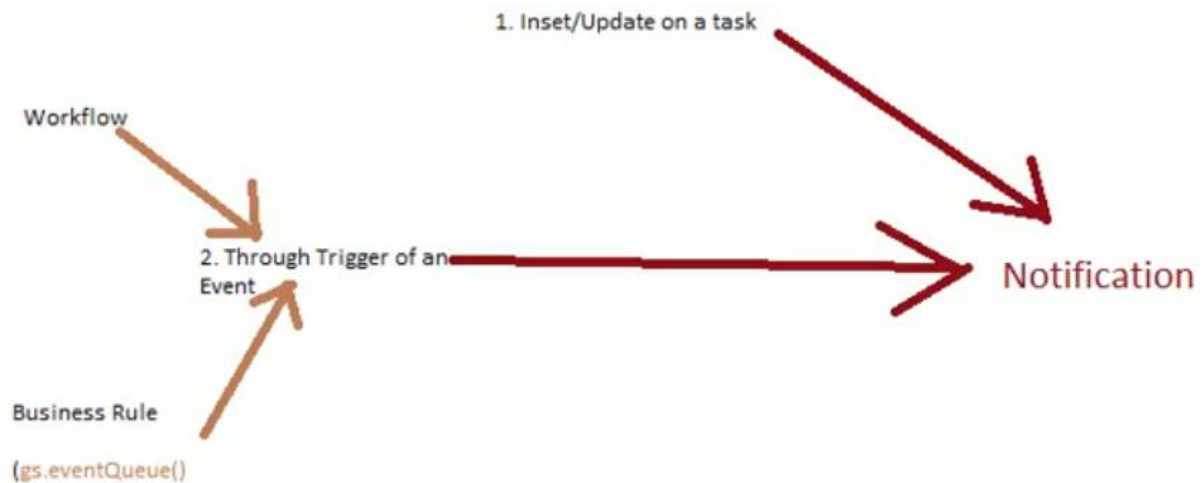


tony.barratt • 2y ago

Hi Jai,

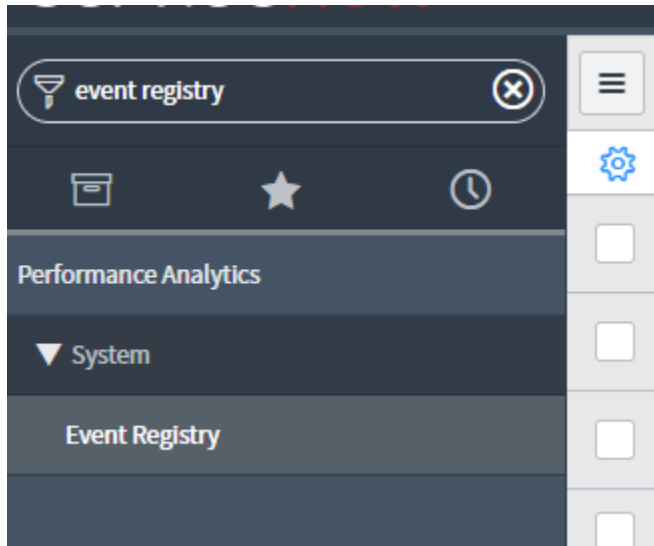
You can define when to send a notification based on the type of triggering event. The instance can send a notification in either of these circumstances.

- When a record is inserted or updated
- When a specific event runs

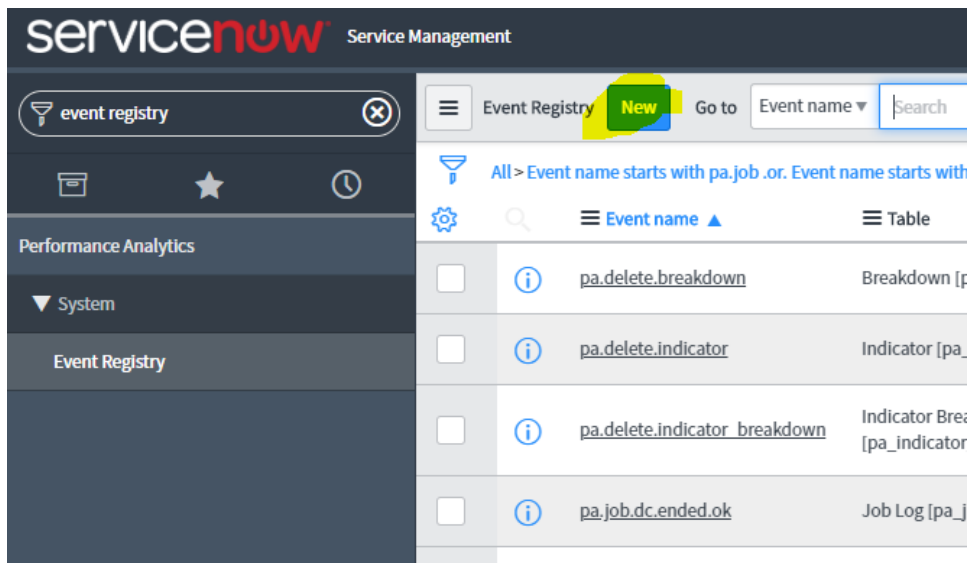


Let's configure a Notification triggered by an **Event** in an Incident:

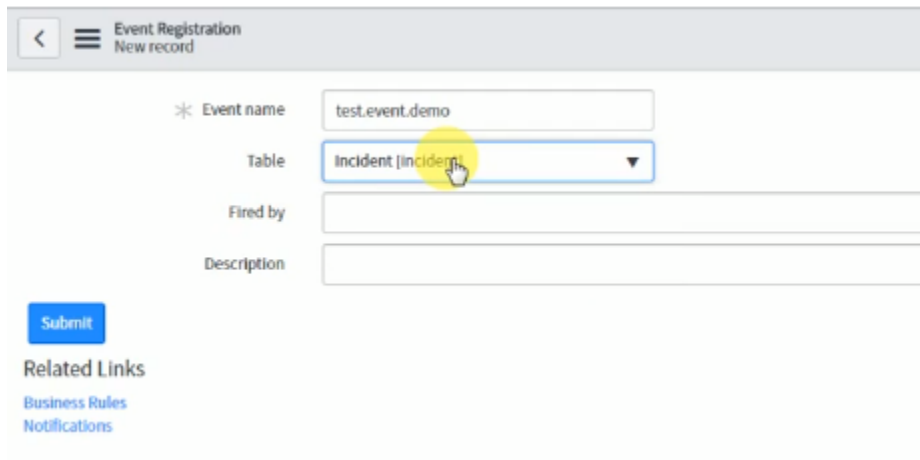
Step One: Go to Event Registry...



...and create a New Event Registry:

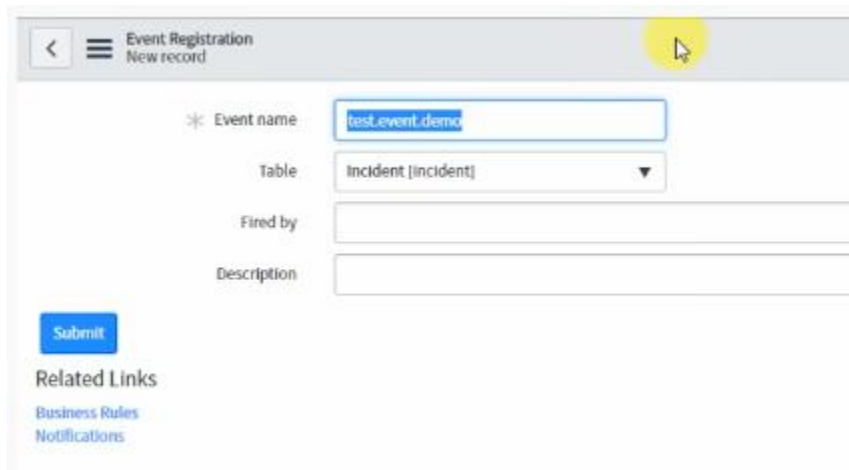


Step Two: Name it and select a table (in this case, we chose incidents):



The screenshot shows a web form titled "Event Registration" with a subtitle "New record". The form has a header bar with a back arrow and a menu icon. Below the header, there are four input fields: "Event name" (containing "test.event.demo"), "Table" (a dropdown menu with "Incident (incident)" selected), "Fired by", and "Description". A blue "Submit" button is located below the "Event name" field. At the bottom left, there is a "Related Links" section with two links: "Business Rules" and "Notifications".

Step Three: Copy the name of the Event (test.event.demo):



This screenshot is identical to the previous one, but with a yellow circle highlighting the "Event name" field, which contains the text "test.event.demo". The "Table" dropdown is still set to "Incident (incident)".

Save or Submit.

Step Three: Open the Notification we recently configured and scroll down to the **When to Send** Tab...

...and change **Send when** to Event is Fired:

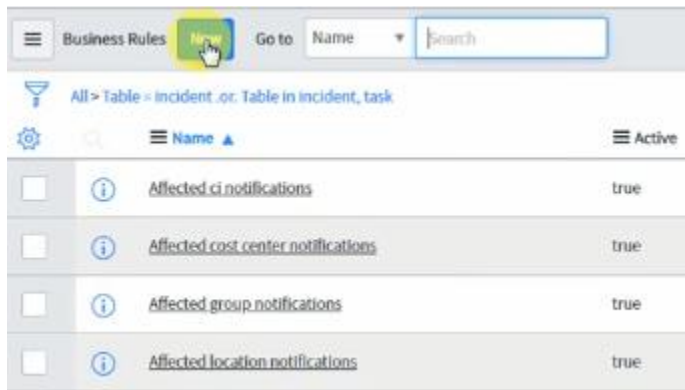
The screenshot shows the 'Notification Test 1 [Advanced view]' configuration page. The 'When to send' tab is selected. The 'Send when' dropdown menu is open, showing options: 'Record inserted or updated', 'Record inserted or updated', and 'Event is fired'. The 'Event is fired' option is highlighted with a yellow circle. The 'Weight' field is set to 0. The 'Updated' checkbox is unchecked.

...now it asks us to specify the **Event name**, so we choose our newly created Event (test.event.demo):

The screenshot shows the 'Notification Test 1 [Advanced view]' configuration page. The 'When to send' tab is selected. The 'Send when' dropdown menu is open, showing a list of events. The 'test.event.demo' event is highlighted with a yellow circle. The 'Event name' field is set to 'test.event.demo'. The 'Weight' field is empty.

Save or Submit.

Step Four: Create a New Business Rule:



...fill out a **Name**, select **Table** (in our case it's Incident), and check the **Advanced** checkbox...

The screenshot shows the 'New Business Rule' form. The 'Name' field is filled with 'test'. The 'Table' dropdown is set to 'Incident [incident]'. The 'Advanced' checkbox is checked. The 'When to run' tab is selected, showing options for 'When' and 'Order'.

Business Rule
New record

A business rule is a server-side script that runs when a record is displayed, inserted, deleted, or when a table is queried. Use business rules to automatically change values in form fields.

Name: test
Table: Incident [incident]
Application: Global
Active: ☒
Advanced: ☒

When to run | Actions | Advanced

Specify whether the business rule should run on **Insert** or **Update**. Use **Filter Conditions** to specify under which conditions the business rule should run.

When: before
Order: 100
Insert: ☐
Update: ☐
Delete: ☐

Under the **When to run** tab, check the **Insert** checkbox:

<

≡

Business Rule
New record

A business rule is a server-side script that runs when a record is displayed, inserted, deleted, or when a table is queried. Use business rules to automatically change values in for

Name

test

Table

Incident (incident) ▼

Application

Global

Active

☒

Advanced

☒

When to run

Actions

Advanced

Specify whether the business rule should run on **Insert** or **Update**. Use **Filter Conditions** to specify under which conditions the business rule should run.

When

before ▼

Order

100

Insert

☒

Update

☐

Delete

☐

Query

☐

Filter Conditions

+ Add Filter Condition

+ Add Filter Condition

Under the Advanced tab, write out the script:

```
(function executeRule(current, previous) {  
    gs.eventQueue("test.event.demo", current, gs.getUser(), gs.getUserID());  
})(current, previous);
```

The screenshot shows the ServiceNow Business Rule configuration page. At the top, there's a header with a back arrow, a menu icon, and the text "Business Rule" and "New record". Below this is a light blue informational banner that reads: "A business rule is a server-side script that runs when a record is displayed, inserted, deleted, or when a table is queried. Use business rules to automatically change values".

The main configuration area has two columns. The left column contains a "Name" field with the value "test" and a "Table" dropdown menu currently set to "Incident [Incident]". The right column contains an "Application" dropdown set to "Global", and two checkboxes: "Active" (checked) and "Advanced" (checked).

Below the configuration fields are three tabs: "When to run", "Actions", and "Advanced". The "Advanced" tab is selected. It contains a "Condition" field (empty) and a "Script" field. The "Script" field has a toolbar with various icons (undo, redo, search, etc.) and a text area containing the following JavaScript code:

```
1 (function executeRule(current, previous /*null when async*/) {  
2  
3     gs.eventQueue("test.event.demo", current , gs.getUser() ,gs.getUserID());  
4  
5 })(current, previous);
```

Save or Submit.

...so now, when a new Incident is created, it will be based on the Event we configured!