

Add Custom Events to Audit Log

Last Modified on 12/02/2024 1:47 am EST

V10.7

Overview

The case audit logs capture details of the events that occurred on a case, records the event details in the portal, and saves them to the *tblLog* table in the database.

The case Audit log by default displays the system events details only and doesn't display the custom events. However, with an *Insert to Audit Log* built-in command activity you, as a workflow developer, can add the custom event details to the audit log, which is then updated in the *tblLog* table.

Procedure

1. In the Admin site, open the workflow in which you want to add the built-in command.
2. In the workflow, add the built-in command activity.
3. In Process Operations category, select *Insert to Audit Log*.
4. Click **Next**.
5. Add Input parameter bindings.

Insert to Audit Log - Input Parameter Bindings

Request

- InsertToAuditLog
 - long workflowInstanceId = 0
 - long masterWorkflowInstanceId = 0
 - int performedByUserId = 0
 - int performedByOriginalUserId = 0
 - string eventName = null
 - string eventDescription = null
 - DateTime date

General

(Name) **InsertToAuditLog**

IsOneWay

Security

Credential Type **None**

Credential Name

Back **Finish** **Cancel**

6. Click **Finish**.

Access case audit log in the portal

You can access a case audit log in the portal in multiple ways.

- Click the Audit Log tab in the Last Message quick access of a case.
- Click the View Audit Log in the vertical ellipses for a case in the portal grid.
- Click the Audit Log in the vertical ellipses in the case details page.
- View case audit log customized in a form. This option is available only when a workflow developer configures an audit log control in a form.

For details on adding audit log in a form, see [this article](#).

Tips

1. In the process grid (commandManagerId: sq.ui.commands.ft.instances), show a Context Menu Item for a specific workflow template:

```
displayRules:  
- type: Rule  
  value: ': ToString($item.WorkflowId) == "b2b6f7ae-914d-4cf7-bc8e-475d0b0a158a" '
```

2. In the process grid (commandManagerId: sq.ui.commands.ft.instances), show a Toolbar Item for a specific workflow space:

```
displayRules:  
- type: Rule  
  value: ': ToString(rt.HttpRequest.Headers["workflowspaceid"]) == "2858595b-14a6-4557-bdc9-eba4e6  
cbf881" '
```