Custom Audit

# Overview

A solution that will allow you to create rows in a Custom Audit table for enhanced analysis. This solution consists of a custom .NET plugin and a model driven app to configure and monitor the auditing.

Disclaimer: This solution is authored by employees of Microsoft but is in no way endorsed by Microsoft Corporation. Code included is custom and can be modified by the receiving user and therefore is NOT supported by Microsoft or any of its affiliates or partners. No warranties or support are expressed or implied.

# Prerequisites

In order to audit a table, the table MUST be enabled for **Creating a new activity**. This can be enabled by editing the properties of the table in the Power Apps maker tool.

**Note: Ensure you are in the correct environment at the top right of the screen.**

1. Select **Tables** in the left navigation.
2. Locate the table you wish to audit and click on the name.
3. Choose **Properties** to open the table properties screen.
4. Click **Advanced options** to expand the options.
5. Scroll to the **Make this table an option when** section and ensure that the **Creating a new activity** is checked.
6. Click **Save** to update the properties.

# Contents of Solution

After importing the solution, a number of items are now available in your environment.

1. A new Activity type table is created called **Custom Audit**. This table will store data in these columns.
   1. **Table Name**: Friendly name of the table the audit row applies to
   2. **Column Name**: Friendly name of the column the audit row applies to
   3. **Changed On**: This is the standard *Created On* column that is renamed to Changed On
   4. **Changed By**: This is the standard *Owner* column that is renamed to Changed By
   5. **Action**: This is the action message sent by the plugin. Should be either **Create** or **Update**
   6. **Regarding**: This is the standard *Regarding* column for an activity row. It links back to the original record that was being audited.
   7. **Old Value**: A text column that stores the old value of the audited row BEFORE the change.
   8. **New Value**: A text column that stores the new value of the audited row AFTER the change.
      1. Note: Old Value and New Value are limited to 4,000 characters. If you need to store more data, you will have to customize the table with a larger multi-line text column. You will also need to customize the .NET plugin to add the data to the new column.
2. Model Driven App – Custom Audit
   1. This app is where you can configure the plugin using a custom HTML/JS Web Resource.
3. Plugin-in assembly – MSFTPP.Plugin.CustomAudit
   1. This is the custom .NET Plugin that handles the auditing of the configured tables and logs the data to the Custom Audit table.
4. Web Resources
   1. msftpp\_CustomAuditBuilder.html – HTML Front end UI for configuring the Audit Steps
   2. msftpp\_CustomAuditBuilder.js – JS backend code that works with the HTML front end
   3. CustomAudit.svg – SVG image for the Custom Audit table

# Configuring a table for audit

Once you have confirmed that the table you wish to audit is enabled for activities, you can now configure which columns will be audited.

1. Launch the Power Apps maker tool and select the environment.
2. From the Left Navigation, select Apps
3. Locate the Model driven app called **Custom Audit Builder** andClick to Play this app.
4. From the Left Navigation, select **Builder**.

## Adding a new Table/Message for audit

1. To Add a New Table and Message (create/update) for auditing, click **+ Add New** button.
   1. You will be presented with a list of tables. If the table you want to audit is not shown, recheck that the table has been enabled for activities. Only tables that are enabled for activities will show in the list.
2. Select the Table to Audit
3. Select the Message of **Create** or **Update**.
   1. The **Create** message will audit when a new row is added to the table and put all of the New Values for columns you request to be audited. This can be used as a baseline set of audit rows for reporting.
   2. The **Update** message will audit when a row is updated with Old and New Values. If a column did not previously have a value (e.g. is was NULL), then the Old Value will show as blank and the New Value will contain what was added.
4. Once you select a message, you will be presented with a list of all auditable columns in the table. Select the checkbox next to the columns you wish to audit.
   1. Note: Some standard columns are not presented (e.g. Created On, Modified On, etc) as they should not be audited.
   2. As a best practice it is recommended that you DO NOT select ALL columns to be audited. Be selective about what really needs to be logged.
5. After selecting the columns, click the **Submit** button at the top right side of the page. This will create the appropriate Steps and Images required in the plugin registration to audit the table.

## Edit and Existing Plugin Step

1. If you have previously configured a plugin step to audit a table and wish to modify that step (e.g. add/remove columns), you can select the **Edit Existing** button**.**
2. You will be presented with a list of the previously configured steps for the Audit plugin. Select the step you wish to modify.
3. A list of columns is built showing ones that have already been selected for audit. Add or Remove columns by checking or unchecking the box next to the column name.
4. After selecting the columns, click the **Submit** button at the top right side of the page. This will update the Plugin and make the changes.

## Manual Plugin Registration Steps to add a table for auditing

1. Locate and launch the **XRMToolbox** and Connect to the environment you wish to update.
2. Load the **Plugin Registration** plugin
3. In the Tree, locate the **(Assembly) MSFTPP.Plugin.CustomAudit** and expand it.
4. Expand the **(Plugin) MSFTPP.Plugin.CustomAudit** node.
5. Right Click on the **(Plugin) MSFTPP.Plugin.CustomAudit** node and select **Register New Step**
   1. **Message**: This will either be **Update** or **Create**, depending on which action you are auditing.
   2. **Primary Entity**: This will be the logical/sdk name for the entity being audited.
      1. e.g. account
   3. **Secondary Entity**: Leave this blank
   4. **Filtering Attributes**: This is where you select which columns being updated will fire the plugin. It is recommended that you click the ellipsis button, deselect all, and then select ONLY the columns that you wish to audit from the list.
   5. Event Handler, Name, Run in Users Context, Execution Order should be defaults
   6. **Eventing Pipeline Stage**: Post-operation
   7. **Execution Mode**: Asynchronous
      1. Also check the box next to **Delete AsyncOperation if StatusCode = Successful**
   8. **Optional**: In the Unsecure Configuration box, you can put **traceon=true** to help trace issues in the plugin trace log. In production you can leave this out, or set to **traceon=false**
   9. Click **Register new step** to save your changes.
6. Right-Click on the new (Step) you just created and select **Register New Image**
   1. **Image Type**:
      1. If this is an Update message, select both Pre Image and Post Image
      2. If this is a Create message, only Post Image should be available
   2. **Name**: Type the word **Target** (case sensitive)
   3. **Entity Alias**: Type the word **Target** (case sensitive)
   4. **Parameters**: Click the Ellipsis to access a list of columns. This is where you select which columns will be passed in for auditing. It is recommended that you click the ellipsis button, deselect all, and then select ONLY the columns that you wish to audit from the list. If you want a column audited, it MUST be in this list. Ideally, this list should match the Filtering attributes you selected in a previous step.
   5. Click **Register Step** to save the changes.