# Actions

Actions provide a large range of critical automation functions in Agiloft, such as sending emails, updating records, importing/exporting data, and managing document requirements like OCR and document comparison. Actions are designed for use in other components such as Rules and Action Buttons, or within other actions, like If-Then-Else actions. Combining multiple actions and conditions strategically makes it easy to carry out complex system operations.

# Creating and Managing Actions

You can create and manage actions from several places in the system:

- In the Actions tab of the Table wizard
- In the Action tab of the Rule wizard
- In the General tab of the Action Button wizard, select **Execute Actions > Add Action**
- From **Setup > Rules**, edit an existing rule or create a new rule, and then click the Action tab
- In the If-Then-Else action wizard, click Add Action

From the Actions screen, you can create a new action by clicking the button for the action type. To edit an existing action, select it in the Actions Library at the bottom of the screen and click Edit.

## Actions Library

Once actions are created and stored in the Actions Library, you can reuse them in other rules or actions in the same table, or through linked record actions from other tables.

In the Actions Library, action names are prefixed with a letter denoting the action type. The `ext` prefix is used for multiple different actions.

For more information about specific action types, see:

- Action Buttons
- Browser Pop-up Actions
- Data Conversion Actions
- Delete Actions
- Document Actions
- Document Analysis Action
- If-Then-Else Actions
- Import and Export Actions
- Integration Actions

- Linked Record Actions
- Machine Learning Actions
- Modal Actions
- Notification Actions
- Publish Actions
- Report Actions
- Script Actions
- Update Fields Actions
- Validate Actions
- Webhook Actions

# Naming Actions

When creating new actions in your system, it is important to follow a consistent naming convention. Use enough detail to make it easy for another administrator to find the action and identify its purpose from the name. For example, some of the default system actions use the following naming conventions:

## Email Action

`Email [recipients] of [condition]`

**Example**

- Email Contract Manager Team of New Contract
- Email Contract Owner of Upcoming Ins Cert Expiration

## Update Fields Action

`Set [field] to [value] or Update [field]`

**Example**

- Set Contract Status to Active from DS
- Set Alert Color to Default
- Update Demo Dates

## If-Then-Else Action

`[Description of collection of conditions]`

**Example**

- Create Actions for Document Approvals
- All Change Approval Edit Actions

# Action Permissions

Actions act with admin-level permissions and override user settings, so make sure to restrict access to the component used to run the action. For example, use visibility conditions to hide Action Buttons from users who shouldn't run the attached actions. Design guard rails to prevent users from running actions accidentally or inappropriately.

For more information, see Action Button Permissions.