

Rules

A rule is a statement of automated logic that the system executes to perform an action. For example, these are some typical rules:

- When a new Lead record is created, email the sales manager.
- When a Contract is created, automatically set the renewal date to one year from now.
- When a user selects a workflow in a Service Request, automatically create Task records linked to that workflow.

Rules are used to create a powerful system that can automatically perform multiple actions. With rules, you can automate email notifications, lead users through appropriate workflows and processes, populate field values, assign records to individual users or teams, generate new records or delete existing records, run scheduled imports and exports, and more.

Rules are triggered by different conditions, and these conditions are highly customizable. For example, one rule might email the contract manager whenever someone creates a new contract. In contrast, another rule might delete thousands of old support cases and runs only once every 30 days during non-working hours.

You can choose which records a rule operates on with a [saved search](#). This focuses the rule so it only processes relevant records. For example, instead of running a data validation rule on every record in the table, you might run it only on records where the relevant data was updated.

Rule Types

Agiloft uses three different types of rules. These rules are categorized by how they are triggered in the system. There are event-specific rules, time-based rules, and summary condition rules.

Event-specific Rules

Event-specific rules are the most common type of rule and run when a record is created or edited. In practice, the "Deleted" option is almost never used. Event-specific rule names should begin with either Create, Edit, or Create/Edit depending on which event causes them to trigger. Some examples of event-specific rules are:

- When a customer creates a Support record, automatically assign it to the Support team.
- When a Contract record's Notes field is edited, email the contract manager.

Apply Rule:

When a Contract is

☒ Created

☒ Edited

☐ Include edits made by other rules during record creation

☐ Deleted

Only when the change occurs by means of:

☒ Email

☒ Web

☐ API

Apply Rule options for Created, Edited, Deleted, and method (Email, Web, or API)

The "Only when the change occurs by means of:" section is used to limit the system interactions that cause the rule to run.

- **Email:** A change is made by an **inbound email**. This is different than sending an employee an email template with a hotlink to edit or immediately change a record.
- **Web:** A user interacts with a record on the Agiloft interface. An example would be a user manually editing the Status field of a Contract record. Web also refers to when a **Web Service** call causes a change.
- **API:** A rule or action makes a change. For example, consider a system with two rules: one that updates a field from Pending to Abandoned after 14 days, and one that emails the manager whenever a record is marked as Abandoned. You can prevent the manager from getting emailed by unchecking the API option of the rule that emails the manager.

For example, if a rule should send an email to a contract manager whenever the contract is updated by a user, select Edited and Web. If a rule should send an email to an accountant whenever a new Travel Expense record is created by a rule, select Created and API. If a rule should send an email to a developer whenever a new Developer Task record is created or edited by a user or rule, select Created, Edited, Web, and API. This is one of the first areas to consult when troubleshooting an event-specific rule.

Time-based Rules

Time-based rules are triggered at specified time intervals, such as every four hours, once a day, or every Monday. They're often used for automating escalations and notifications, as well as scheduling resource-intensive actions like imports during non-business hours. Consider these time-based rules:

- Every night at 10 PM, import data into the Leads table to create new records.

- Notify the support manager once a day of all open cases that have critical severity and were created more than 1 day ago.
- Notify the assigned person that the due date for a task is today.

Time-based rules can be triggered at any time of day, but their maximum frequency depends on how your knowledgebase is hosted. If your knowledgebase is hosted on Agiloft servers, time-based rules can't run more often than every 5 minutes. If you use your own server to host your knowledgebase, you can shorten the minimum frequency on the Knowledgebases tab of the [admin console](#).

Apply Rule


☒ Every 1 ▼ hours ▼

☐ Weekly on Monday ▼

☐ Monthly on 1 ▼ day of the month

Time-based schedule

If an error occurs during a time-based rule, an email is automatically sent to members of the admin team and any child teams.


 For troubleshooting purposes, it's important to be able to tell if an email is generated by a time-based rule or an email is generated by a scheduled report. For more information on reports, visit the [Reporting page](#) or the [Schedule](#) section of the Create and Edit Charts and Reports page.

Summary Condition Rules

Summary condition rules are triggered based on the number of records in a table that meet a specified condition at a specified time interval. Consider these summary condition rules:

- Notify the Sales Manager if the number of Leads expected to close next month is less than 20 or more than 40.
- Email the contract manager every Monday if the contracts expiring this month are valued at more than \$100,000.

Summary condition rules are the only rule type that actually requires a saved search during configuration.

 By default, only one time-based or summary condition rule is allowed to run in a knowledgebase at any one time. If you host Agiloft on your own server, you can modify this by customizing the [Max Timer Rules per KB](#) global variable.

Creating and Editing Rules

Rules are created and edited with the Rules wizard. To access the Rules wizard, click **Setup [Table]** on the left pane and click the Rules tab. In the list of rules, click **New** to create a new rule, or click the edit icon next to an existing rule. Alternatively, you can access all the rules in your system by clicking the **Setup** gear in the top-right corner going to Rules.

General Tab

Use the General tab to configure basic settings for the rule, such as the rule's name and how the rule generates history entries.

1. Provide a description for the rule. Be as descriptive as possible so that another admin can look at the rule and know its purpose. For more information on naming rules, see [Organizing and Naming Rules](#).
2. If you started with **Setup > Rules** instead of **Setup [Table]**, select the table in which the rule will run. If you accessed the Rules wizard from the Table wizard, the current table is automatically selected.
3. Select whether the rule is applied to any subtables. This refers to any existing subtables and any subtables that are added after the rule is created.
4. Select whether the rule is enabled.

✔ If you're configuring rules in a system that is not yet live, it's often a good idea to disable time-based rules until the system goes live.

5. Select how you want the rule to generate history entries:
 - **Always record history entries:** The rule generates history entries whenever it runs.
 - **Only record history entries when record is modified by actions called by the rule:** The rule only generates history entries when it modifies records with the actions it contains.
 - **Do not record history entries:** The rule never generates history entries, even if it modifies records.

✔ Test the potential impact of the rule and come back to this section later to properly determine how the rule should produce history entries. If you are using a time-based rule that runs on frequent intervals or as a series of nested rules, consider using the "Only record history entries when record is modified by actions called by the rule" option. This way, the History tab stays clean and decluttered. Huge history tables can increase backup time, complicate troubleshooting, and cause other performance issues.

6. If you're working with a time-based or summary condition rule, click **Run the rule now** to **test the rule** without waiting for the scheduled run time. If you've just created a record or made an edit in an existing record, this button reads **Save & Run Now**.

7. Click Next.

Rule Type Tab

Select the rule type here. The remaining tabs of the Rules wizard differ based on the selection made here.

Select the type of rule:

- **When a [record type] is created/edited:** Creates an event-specific rule. If you select this type of rule, you have two additional options:
 - **Run in background:** Forces the rule to run after all other triggered rules have completed their changes. If a rule depends on changes made by other rules, use this option. If a [system is running slowly](#), check your rules and make sure Run in background is selected for rules that impact a lot of records.
 - **Disable loop protection:** Turns off the loop protection mechanism for rules that run in the background, which means that rules that preceded or triggered the current rule can be re-triggered by the current rule. This creates the potential for infinite loops. Disable Loop Protection should almost never be selected because loops can cause serious performance issues, and only becomes available when Run in background is selected.
- **At selected time intervals:** Creates a time-based rule.
- **When some summary condition based on multiple has/hasn't been met:** Creates a summary condition rule.
- Click Next.

Condition Tab

Use the Condition tab to determine when the rule runs. The first section of the Condition tab changes depending on the decision in the Rule Type tab. The "When saving a record" and "Condition" areas of the Condition tab are always present, regardless of rule type. However, for event-specific rules, the tab is referred to as Search Condition.



By default, you're notified before running a rule that matches more than 100,000 records. If you choose to run such a rule, it only affects the first 100,000 matching records. You can change this behavior at the [admin console](#) or KB level by setting new values for the following global variables:

- Max Rule Records Limit
- Max Rule Records Warning

Event-specific Rules

1. Choose whether the rule is triggered when a record is created, edited, or both. If Edited is selected, the option to select "Include edits made by other rules during record creation" becomes active. Select this option if you want an edit made by a rule on a newly created record to trigger the new rule, or if you want your rule to run when a rule changes a value in a newly created record.
2. Select what interfaces can trigger the rule: Email, Web, or API.
3. Select what happens after the rule runs, when the impacted records are saved:
 - **Run rules:** Allows other rules to be triggered by the changes. If the rule shouldn't trigger additional rules, make sure Run Rules is not selected.
 - **Update defaults:** Updates default values in the records, such as the Date Modified and Last Updated By fields. These values are automatically updated with values referring to who and when made the last edit on a record.
4. If desired, select or create a saved search that the records must meet for the rule to be triggered. For example, if you have a rule that assigns certain change requests to the change manager, you might use a saved search to specify that the Urgency field be set to Emergency for the rule to be triggered.
5. Click Next.

This concludes the Condition tab section for event-specific rules. Feel free to skip ahead to the [Schedule tab](#) if you do not require information on time-based or summary condition rules.

Time-based Rules

1. Choose whether the rule runs on every record or only on records that meet the saved search condition. It's important to make sure you choose correctly here, as an incorrect choice can cause a rule to run thousands of times instead of just once.
2. Select what happens after the rule runs, when the impacted records are saved:
 - **Run rules:** Allows other rules to be triggered by the changes. If the rule shouldn't trigger additional rules, make sure Run Rules is not selected.
 - **Update defaults:** Updates default values in the records, such as the Date Modified and Last Updated By fields. These values are automatically updated with values referring to who and when made the last edit on a record.
3. Choose what to do if an error is encountered while updating records:
 - **Continue with remaining records:** The rule continues to process the rest of the records. Note that this can increase the total time it takes the rule to make any updates.
 - **Stop rule immediately:** The rule stops at the first error and does not process additional records.
4. Choose whether you want an error report sent to the admin team, if an error is encountered.

5. If desired, select or create a saved search that the records must meet for the rule to be triggered. For example, if you have a rule that assigns certain change requests to the change manager, you might use a saved search to specify that the Urgency field be set to Emergency for the rule to be triggered.
6. Click Next.

This concludes the Condition tab section for event-specific rules. Feel free to skip ahead to the [Schedule tab](#) if you do not require information on summary condition rules.

Summary Condition Rules

1. Choose whether the rule runs once if the search is met or once for each record found by the search.

Options
☐ Execute Actions once if the search is met
☒ Execute Actions once for each record found by search

Summary condition options

2. Select what happens after the rule runs, when the impacted records are saved:
 - **Run rules:** Allows other rules to be triggered by the changes. If the rule shouldn't trigger additional rules, make sure Run Rules is not selected.
 - **Update defaults:** Updates default values in the records, such as the Date Modified and Last Updated By fields. These values are automatically updated with values referring to who and when made the last edit on a record.
3. Select or create a saved search that identifies what types of records to count towards the condition. For example, if you have a rule that sends an email to a manager if 15 tasks are marked as complete, you'd create a saved search that targets records with a Status of Complete.
4. In the Numbers section, select how many records the saved search condition must find for the rule to be triggered. For example, if you have a rule that sends you an email when you complete 15 or more tasks in a week, set this value to 15.

Number
greater or equal, >= ▼ 15

5. In the Group Data By section, select a field for grouping the data.
6. Click Next.

Schedule Tab

Use the Schedule tab to set when the rule is active and its order of priority relative to other rules in the table. All rule types include both the Rule is Active and Priority sections. Time-based and summary condition rules have extra sections called Apply the Rule and Start Time, which are used to determine time intervals. Summary condition rules have a unique section called Frequency.

1. Select whether the rule is always active, active only during the working hours of a desired team, or active only outside the working hours of a desired team. Certain rules, such as rules that run scheduled imports, are typically set to run outside of a team's working hours because they can have a negative impact on system performance.

Rule is Active

☒ Always

☐ Only during working hours of this team:

☐ Only outside of the working hours of this team:

Company Team ▼

Rule is Active options

- ✔ After you finishing configuring the rule in the Rules wizard, it's a good idea to double-check that the team you've chosen has the working hours you expected.

2. For time-based and summary condition rules, select the start time and how often to apply the rule.
3. Set the rule's priority. Rules execute in their priority order relative to other rules in the same table, and rules with the lowest number execute first.

- ✔ Rules that rely on information updated by other rules must run after those rules; a lower priority correlates to a higher value. For instance, if rule A relies on information updated by rule B, you must give rule A a higher value so that it runs after rule B.

4. For summary condition rules, select how frequently the rule should run with the Frequency section.
 - Once: Applies the rule when the condition is met, then does not apply the rule again until the condition is broken and then met a second time. For example, consider a rule that sends a message if the number of open issues exceeds 10. If the frequency is once, the rule sends a message when the number of records reaches 11, but doesn't send any more until the number drops below 11 and then rises again.
 - For as long as the condition remains true: Runs the rule every time the saved search returns a record. If you select "for as long as the condition remains true" for the same rule, the rule sends a message for every record the saved search finds that exceeds 10. You receive a message for record 11, 12, 13, and so on.
5. Click Next.

Action Tab

Use the Action tab to add actions to the rule. Actions are executed in the order listed from top to bottom, but subsequent actions don't use values updated by previous actions. This means that if subsequent actions rely on values updated by previous actions, you must place the subsequent actions in a separate rule that runs after the initial rule is complete. This is necessary because values aren't saved to the database until the rule is completely finished running.

For example, suppose you have a rule that runs an Update Fields action to change the status of a record from Closed to Reopened if the Additional Information field is updated. Let's say you also want to run a Validate action that notifies the user that they can call a support line if the priority is Critical when the status changes to Reopened. If you don't place these actions in separate rules, the system won't recognize the change to Reopened, and the Validate action won't run, even if the record's priority is Critical.



It's a good idea to put all [Validate actions](#) in a rule that runs before any others. This prevents other actions from making changes before any validations can run.

1. Click one of the buttons to create a new action, or select an action from the Available Actions pane and move it to the Selected Actions pane.
2. Determine the order of the selection actions by selecting the action and clicking either Move Up or Move Down.
3. Click Finish to create the rule.



[Email actions](#) run in the background by default. This means that they might not complete before the next action. For example, if you have an Email action followed by an Update Fields action, the fields might be updated before the email sends, making the email unexpectedly include the updated field values. You can choose to run an Email action in the foreground in the "Background processing type" section of the Email Action wizard. This makes sure it's sent before any of the actions that follow.

Running Rules

When you save a record and trigger rules, various system actions occur in a regimented order. For example, rules that are executed in correlation to their [priority order](#) is one such system action. It's important to know when a rule runs, relative to other system actions. This allows you to better anticipate how the system behaves when a user saves a record.

When a record is saved, system actions are executed in this order:

1. Required fields and any other fields with their own data type validations are evaluated first. For example, the system might check if an Email field contains an "@" character.

2. Dynamic default values are updated. For example, the system might populate the Updated By field with the user making the update and Date Updated field with the current date, automatically.
3. Rules are executed in their priority order:
 - a. If a rule contains Validate actions, it can prevent the record from being saved, as well as prevent subsequent rules and system actions from being executed. For example, a Validate action might prevent the record from being saved if a certain field hasn't been updated.
 - b. If a rule updates one or more fields in the record, rules triggered by API changes are run again, in order.
 - c. The next rules in the priority order are executed.
 - d. **Background rules** are executed after all other rules are complete.
4. Workflow actions, if any, are executed.
5. Scripts associated with a rule or workflow action are executed.



Interactions between rules, priority order, and order of execution are important variables to consider when you design rules. Rules can overlap, invalidate one another, trigger other rules, run on incorrect values, and cause other problems if you don't consider these factors during the design process. To solve existing issues with rules, consider checking out [Troubleshooting Rules](#).

Disabling Rules

Usually, it's better to disable a rule instead of deleting it, in case you need the rule in the future. If you're configuring rules in a system that is not yet live, it's also a good idea to disable time-based rules until the system goes live.

To disable a rule:

1. Go to **Setup > Rules**, or click Setup [Table] on the left pane and click the Rules tab.
2. Click the edit icon next to the desired rule.
3. On the General tab of the Rules wizard, add "Disabled:" to the beginning of the rule description so that it's clear the rule is disabled.
4. Scroll down and select No for Rule is enabled.
5. Click Finish. The rule will no longer run, but it's retained in the system if you need to enable it again in the future.