

# Troubleshooting Rules

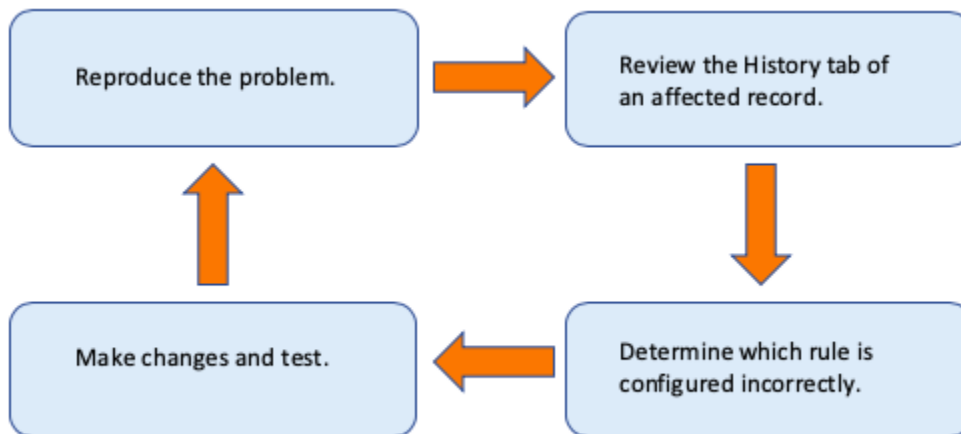
---

A rule is a statement of automated logic the system executes in order to perform an [action](#) or multiple actions. Rules are a lot like action buttons, but instead of clicking a button to trigger them, they run automatically based on preconfigured settings. Most systems accumulate increasingly complex rules as they become more robust. The first several sections include details about each stage of the troubleshooting process. To skip to examples that show the complete process, go to the [Troubleshooting Examples](#) section of this page.

✔ Admins can prevent many issues by following [best practices](#) when initially configuring a rule. Familiarity with best practices makes it easier to spot and diagnose problems with rules.

The general troubleshooting process for rules is:

- Notice an issue and document the steps needed to reproduce it.
- Determine which rule is causing the issue.
- Check the configuration of the rule.
- Make changes and then test the rule.



## Recognize the Source of the Problem

---


Before any troubleshooting the source of the issue must be determined. In general, rules are responsible for:

- Problems where an action fails to trigger an automatic response,
- An action triggers an unexpected response.
- Or some automatic process no longer runs on time or as expected.

For example, if a table isn't showing in the left pane, it's safe to say no rules require troubleshooting because rules don't control the left pane. If creating a new employee record is supposed to automatically email the new employee's manager, but that manager doesn't receive an email or receives the incorrect email template, the system almost certainly has an issue with an event-specific rule.

When a problem arises, it's vital to take note of what caused the issue, and to reproduce the steps if possible. This leads to a simpler testing process, and can help determine if a rule is the issue. If a rule is in fact the issue, knowing what caused it to run can help identify the specific rule.



The best place to learn more about an issue is by checking a record's History tab. Check the History tab of records that have been improperly updated by a rule, or records that did not get updated when they should have.

 If a record wasn't updated, it doesn't necessarily mean the rule didn't run. For example, a condition in a rule's If-Then-Else action could be met that doesn't make any changes to the record. In this case, the behavior of the rule is still included in History Event Details.

Here are two examples of which records to check given some common circumstances:

- If the system is supposed to send an email to a new employee's manager when a new Employee record is created, but the manager doesn't receive an email, it's likely there's an issue with an event-specific rule in the Employee table. Check the History tab of the new Employee record.
- If the system is supposed to update a Contract record's Status to Rejected if it isn't set to Approved after 21 days, but the Status was set to Rejected after two days, it's likely there's an issue with a time-based rule in the Contract table. Check the History tab of any Contract record that was incorrectly updated after two days.

After opening a record, go to the History tab. History Events shows a list of changes that have occurred to a record over time and what caused them, making it an extremely valuable section for testing purposes. The Modified By column tracks who or what made the change, including the names of individual rules that made changes. If the value under Modified By is not a rule, it's still possible a rule is involved. For example, it's possible that a rule ran after someone made an edit. If the Modified Fields value matches the fields that were incorrectly updated in a record, check History Event Details by clicking the magnifying glass icon.

History Events			
	Date	Modified By	Modified Fields
	Jan 26 2021 11:55	Rule: Edit: All Contract Approval Edit Actions (web, API)	Updated By, Number of Approvals Completed, Updater Team, Date Updated, Updater Login, Approved, Pending Approval
	Jan 26 2021 11:53	contractmgr	Date Updated, Times Sent for Approval, Approvals Launched

History Events list

History Event Details contains a detailed snapshot of the actions called by a rule for a specific record. If an action looks incorrect, consult the configuration in the rule's Action tab. Although the format of History Event Details varies depending on the type of action the rule called, the nature of the information stays largely consistent. Consider the History Event Details for the rule modification showed in the History Events example above. The section shows the fields that changed, the before and after values, and the name of the Update Fields action that caused it.

Jan 26 2021 11:55, Triggered by the modification in <a href="#">Approval 464</a>			
Changes made by Rule: Edit: All Contract Approval Edit Actions (web, API)			<a href="#">View Contract After Change</a>
Field Changed	From	To	By
Updater Login	contractmgr	admin	Defaults
Updated By	Robert Barash	Agiloft Admin	Defaults
Approved	--undefined--	1	
Pending Approval	1	--undefined--	
Number of Approvals Completed	0	1	Increment Number of Approvals Received
Updater Team	Contract Management Team	Admin Team	Defaults
Date Updated	Jan 26 2021 11:53	Jan 26 2021 11:55	Defaults
Rule: Edit: All Contract Approval Edit Actions (web, API)			<a href="#">View Contract After Change</a>
Action Name	Action Report		
Increment Number of Approvals Received	Fields have been modified:		
	Field	From	To
	Number of Approvals Completed	0	1

### Example History Event Details

For another example, see the History Event Details below for a record created by an admin that caused two event-specific rules to call two separate If-Then-Else actions. These were both triggered because a record was created, and for each, History Event Details shows the results of the actions that ran.

Jan 27 2021 10:01			
Changes made by admin			<a href="#">View Location After Change</a>
Field Changed	From	To	By
Location Created			
Rule: Create: Copy state/province to new fields			<a href="#">View Location After Change</a>
Action Name	Action Report		
Set State province	If-then-else flow		
	search matched	-else-	
	action executed	Set State/Province from US State	
Rule: Create: On Creation Actions			<a href="#">View Location After Change</a>
Action Name	Action Report		
All Creation Actions by Web/API	If-then-else flow		
	search matched	Street Address 2=NULL	
	action executed	Set address fields for no street address 2	
Field Changed	From	To	By
Address 1 and 2 Comma Formatted	--undefined--	40 Sun Plaza	All Creation Actions by Web/API
Address 1 and 2 in Block Format	--undefined--	40 Sun Plaza	All Creation Actions by Web/API

### History Event Details second example

Although the format of these History Event Details is different, the information gathered here is similar. Instead of showing a list of updated fields, this History Event Detail section shows the If-then-else flow (ITEF) under the Action Report to monitor exactly how actions within rules interacted with a system. The ITEF is used to diagnose and then solve issues; it shows which clause was verified in a rule's If-Then-Else action, and what additional actions ran because of the verified clause. This information is recorded in the "search matched" and "action executed" rows of the ITEF, respectively.

The three If-Then-Else actions called All Creation Actions by Web/API, Set Address fields, and Determine Tax Exempt Status action show what happens when an `If` or an `ElseIf` clause is verified; the "search matched" row will display the details of the matching clause. The Set State province action shows what happens when an `Else` clause is verified. "Action executed" shows the action name regardless of which type of clause is verified. If any of this appears incorrect, consult the configuration of the action in the rule's Action tab.

If you can't find the rule in History Event Details, ensure that the rule is enabled, check the values in the Conditions column, and verify the saved search is set up to include the correct subset.

## Check Rule Configuration

---

After a potential culprit rule is identified using the History tab, check the settings of the rule using the Rules table for improper configuration. To access the Rules table, click Setup [Table] and go to the Rules tab.

Comment	Conditions	Name of Action(s)	Actions Descriptions	Enabled
Edit: All Edit Validations (web only)	Apply Rule: When a Change Request is Edited	All Edit Validations	if (Status changed from: any value to: 'Completed' last user's modification and Number of Tasks>=1 and Number of Incomplete Tasks>=1) { Validate action There are still incomplete tasks shows a message "These must be completed before you can close this Change Request." ...	Yes

This table makes it easy to skim a rule's settings and quickly identify something that might be causing the problem. The columns have specific uses relative to troubleshooting:

- The **Comment** column displays the name of the rule.
- The **Conditions** column displays when event-specific rules are applied. It also displays which saved search is used in a rule. Use this column to quickly check the saved search.
- The **Name of Action(s)** column displays which actions are included in a rule's automation. The Action Descriptions column includes more detail.
- The **Actions Descriptions** column displays the details of each action included in a rule. Use this column for a detailed account of the first few actions included in the rule.
- The **Enabled** column displays whether a rule is enabled or not.

### Example

The All Edit Validations action is an If-Then-Else action. The Action Description begins with an `if`-statement and then a `Validate` action before trailing off. The remaining actions are shown in the If-Then-Else editor.

Actions Descriptions	<a href="#">Add Action</a> <a href="#">Add If</a> <a href="#">Add Elseif</a> <a href="#">Add Else</a> <a href="#">Add Comments</a> <a href="#">Move Up</a> <a href="#">Move Down</a> <a href="#">Edit</a> <a href="#">Delete</a>
if (Status changed from: any value to: 'Completed' last user's modification and Number of Tasks>=1 and Number of Incomplete Tasks>=1) { Validate action There are still incomplete tasks shows a message "These must be completed before you can close this Change Request."...	<pre>if (Status changed from: any value to: 'Completed' last user's modification and Number of Tasks&gt;=1 and Number of Incomplete Tasks&gt;=1) {     There are still incomplete tasks } else if (Status changed from: 'Pending Change Manager' to: 'In Progress' during record's last modification and Service Approvals Required?='Yes') {     Not allowed to Change Status to In Progress } else if (Status changed from: any value to: 'Completed' last user's modification and Number of Tasks&gt;=1 and Number of Failed Tasks&gt;=1) {     Update Status to Completed with Failures }</pre>

## Check Rule Configuration in the Rules Wizard

Review the settings on each tab of the Rules Wizard to make sure they're configured correctly. Many issues with rules come from missing or misinterpreting a single setting. For information on how to diagnose problems with the Rules wizard, compare the rule's settings with the recommendations in the [Creating and Editing Rules](#) section of the Rules article, and make modifications. It's likely that the issue can be traced to improper configuration in the Conditions tab, the Schedule tab, or in an action in the Action tab.

## Test Rules

Test the system after each modification. If a rule is only tested once after several settings were modified, determining what exactly solved the problem becomes more difficult.

To test a rule:

1. Reproduce the problem. For event-specific rules, trigger the event that activates the rule. For time-based and summary condition rules, click **Save & Run Now** in the General tab.
2. Check the record. If the record was updated as expected, and the rule is now working correctly, close the record and ensure the steps you took to fix the issue are documented. If the issue is still present, proceed to step 3.
3. Check the record's History Events section.
4. Open the details of the most recent History Event. Immediate changes might be noticeable by looking at **Modified By** and **Modified Fields**.
5. Check the different sections of History Event Details. Use that information to determine which Rules wizard tab to inspect.

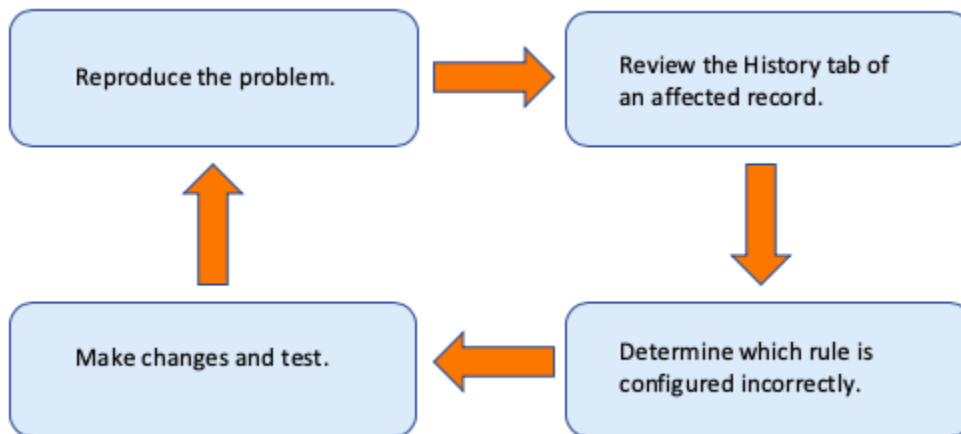
6. Inspect the tabs of the Rules wizard and make a modification. Repeat the process until the rule behaves as desired.

✓ When troubleshooting rules in a non-production environment, change the History entry management value to "Only record history entries when record is modified by actions called by the rule." Changing this option won't reset the History table but will limit what is recorded from now on. This can help declutter the History Events section and make troubleshooting rules a bit simpler. This setting isn't recommended in production environments, as it can prevent important History events from being recorded.

## Troubleshooting Examples

---

The following examples demonstrate the end-to-end troubleshooting process laid out in the previous sections of this article. These examples can help you figure out where to begin with issues you find in your own system.



## Troubleshooting an Event-specific Rule using If-Then-Else Flow

Lei, an admin, receives a ticket saying that a Location record was created with a US State field of WA, but the Income Tax Status field was set to Non-Exempt. According to the ticket, when a Location record is created or edited with a US State field value of WA, the Income Tax Status field should automatically be set to Exempt.

Lei can tell from the ticket that the rule is supposed to run when a record is created or when a specific field is edited, making it an event-specific rule. Here are the steps that Lei should follow to successfully troubleshoot this issue, and other issues like it:



action is being called, either the `if`-statement was improperly configured, the action was improperly configured, or the record was not created with WA in the US State field. Lei hypothesizes that the `if`-statement must be the issue.

Jan 27 2021 10:01

Changes made by admin

View Location After Change

Field Changed	From	To	By
Location Created			
Rule: Create: On Creation Actions			
View Location After Change			
Action Name	Action Report		
All Creation Actions by Web/API	If-then-else flow		
	search matched	Street Address 2=NULL	
	action executed	Set address fields for no street address 2	

Rule: Tax Exempt

View Location After Change

Action Name	Action Report
Determine Tax Exempt Status	If-then-else flow
	search matched -else-
	action executed Set Tax Exempt Status to Not Exempt

Field Changed	From	To	By
Income Tax Status	--undefined--	Not Exempt	Determine Tax Exempt Status

History Event Details for Tax Exempt setting

- Open the improperly configured rule.
- Navigate to the Action tab and edit Determine Tax Exempt Status, since Lei hypothesized that the issue is an improperly-configured rule.
- Click the Details tab. At first glance, the action's configuration seem normal. However, Lei realizes that the State or Province field and the State Abbreviation field are not the same field.

[Add Action](#)
[Add If](#)
[Add Elself](#)
[Add Else](#)
[Add Comments](#)

```

if (State or Province='WA') {
    Set Tax Exempt Status to Exempt
} else {
    Set Tax Exempt Status to Not Exempt
}

```

- Edit the `if`-statement.

[Add Action](#)
[Add If](#)
[Add Elself](#)
[Add Else](#)
[Add Comments](#)

```

if (State Abbreviation='WA') {
    Set Tax Exempt Status to Exempt
} else {
    Set Tax Exempt Status to Not Exempt
}

```

- Click Finish in both screens.

10. To test, Lei returns to the table and creates a new record that should trigger the rule. If the `IF`-statement was the cause of the issue, and it is now configured properly, the record should have the correct values. If the value is still incorrect, double-check the rule's configuration, and repeat the process.




▼ Street Address			
Street Address 1	One Microsoft Way	Street Address 2	
City	Redmond	Country	United States
State Abbreviation	WA	Income Tax Status	Exempt

New record showing tax status set correctly

## Troubleshoot a Time-Based Rule

Imagine it's the morning of December 2nd, 2020. Alex the admin receives a ticket detailing that the system is supposed to update the Date Updated and Date Created fields of each record in the Company table on the first of each month at 9 PM, but the records weren't updated yesterday night. Since this edit is supposed to occur at a specific time interval, it's likely that a time-based rule requires troubleshooting. Here are the steps that Alex should follow to successfully troubleshoot this issue, and other issues like it:

1. Open a record in the Company table.
2. Go to the History tab.
3. Check Date, Modified By, and Modified Fields to determine which history event to investigate. In this example, the top choice has Date Updated and Date Created in the Modified Fields column, and it last occurred close to the first of the previous month. Alex chooses this rule to investigate.
  - a. Alex notices that there's also an additional field that was unexpectedly modified by this rule called Original Start Date. Checking History Events and History Event Details thoroughly is a great way to troubleshoot for additional improper configurations that haven't been noticed yet.

▼ History Events		
[Page 1] <a href="#">Next</a> <a href="#">Last</a>		
Date	Modified By	Modified Fields
 Nov 02 2020 22:30	Rule: TB Demo Data Update: Update date fields by one month each month so reports have data	Original Start Date, Date Created, Date Updated
 Jul 21 2020 09:56	ewsystem	No. of Active Insurance Coverages
 Jul 20 2020 12:11	contractreq	No. of Active Insurance Coverages

History Events list with most recent change including Original Start Date field

4. Click the magnifying glass icon to open the relevant event's History Event Details.

Nov 02 2020 22:30			
Changes made by Rule: TB Demo Data Update: Update date fields by one month each month so reports have data			
<a href="#">View Company After Change</a>			
Field Changed	From	To	By
Original Start Date	Jan 24 2010	Feb 24 2010	Update Demo Dates
Date Updated	Oct 21 2020 13:47	Nov 21 2020 13:47	Update Demo Dates
Date Created	Jan 25 2010 07:49	Feb 25 2010 07:49	Update Demo Dates
Rule: TB Demo Data Update: Update date fields by one month each month so reports have data			
<a href="#">View Company After Change</a>			
Action Name	Action Report		
Update Demo Dates	Fields have been modified:		
	Field	From	To
	Date Updated	Oct 21 2020 13:47	Nov 21 2020 13:47
	Date Created	Jan 25 2010 07:49	Feb 25 2010 07:49
	Original Start Date	Jan 24 2010	Feb 24 2010
Field Changed	From	To	By
Original Start Date	Jan 24 2010	Feb 24 2010	Update Demo Dates
Date Updated	Oct 21 2020 13:47	Nov 21 2020 13:47	Update Demo Dates
Date Created	Jan 25 2010 07:49	Feb 25 2010 07:49	Update Demo Dates

#### History Event Details

5. Begin to gather information. In the example, the History Event Details show the fields that changed, the before and after values of those fields, and the action that changed them.
- The only rule listed is "TB: Demo Data Update: Update date fields by one month each months so reports have data." This is the rule that should be investigated back in the People table.
  - The date at the top of the window indicates that the rule caused this event to occur on March 02 2020 22:31. Since this doesn't match the time and date the rule is supposed to run, Alex makes a note to check the Schedule tab of the rule.
  - This rule should only update the Date Updated and Date Created field. However, there is an additional row under the Fields Changed column for Original Start Date. Update Demo Dates is the action listed as updating this field. Alex makes a note to check the configuration of this action in the Actions tab of the rule.
6. Click Setup Company and navigate to the Rules tab.



The following steps show how to troubleshoot the example rule based on information gathered in step 5. Every time-based rule is configured differently, but the process is similar.

7. Open the potentially misconfigured rule.
8. Go to the Schedule tab of the Rules wizard.

- a. In the example, the Schedule tab shows the rule was incorrectly configured to run on the 2nd day of the month at 10:30 PM. Alex adjusts these values to reflect the desired outcome.


Apply Rule

☐ Every 1 hours

☐ Weekly on Monday

☒ Monthly on 2 day of the month

Apply Rule setting on Schedule tab

 Normally a rule should be tested after each adjustment. However, there is no way to test the correct time interval, besides waiting for that time interval to come and go. Luckily, every other aspect of a time-based rule can be tested as soon as it is configured.

- 9. Navigate to the Action tab. Scroll down to the Selected Actions section and edit Update Demo Dates.
- 10. Navigate to the Fields tab. Scroll down to Original Start Date and clear the checkbox.
- 11. Navigate to the Values tab and Click Finish. Since a possible cause was found and fixed, Alex decides to test the rule to make sure it no longer updates the Original Start Date field.
- 12. Navigate to the General tab. Scroll down and click Save & Run Now to test the rule. Save & Run Now can be stylized as Run the rule now if no recent edits were made to the rule. The functionality remains the same.
- 13. Navigate back to the record's History tab to check if the modification worked.
- 14. The Modified Fields column in the image below no longer includes Original Start Date. Since this entry is a result of a test, the date in the date column window correlates to when Save & Run Now was clicked.

Modified By	Modified Fields
Rule: TB Demo Data Update: Update date fields by one month each month so reports have data	Date Updated, Date Created

History Events with Original Start Date not listed

15. Click the magnifying glass. The Original Start Date is no longer included in the Field Changed column here either.

Jan 27 2021 11:15			
Changes made by Rule: TB Demo Data Update: Update date fields by one month each month so reports have data			
<a href="#">View Company After Change</a>			
Field Changed	From	To	By
Date Updated	Dec 21 2020 13:47	Jan 21 2021 13:47	Update Demo Dates
Date Created	Mar 25 2010 07:49	Apr 25 2010 07:49	Update Demo Dates
Rule: TB Demo Data Update: Update date fields by one month each month so reports have data			
<a href="#">View Company After Change</a>			
Action Name	Action Report		
Update Demo Dates	Fields have been modified:		
	Field	From	To
	Date Created	Mar 25 2010 07:49	Apr 25 2010 07:49
	Date Updated	Dec 21 2020 13:47	Jan 21 2021 13:47
Field Changed	From	To	By
Date Updated	Dec 21 2020 13:47	Jan 21 2021 13:47	Update Demo Dates
Date Created	Mar 25 2010 07:49	Apr 25 2010 07:49	Update Demo Dates

### History Event Details

Alex has to wait until next month to ensure the rule runs at the correct time, but was able to diagnose why it had ran at the wrong time and changed it accordingly. However, Alex was able to diagnose and then successfully fix another issue that arose during the troubleshooting process for the first rule. This example shows how to use the History Event Details section to troubleshoot for a specific issue as well as how much additional information you can gather from a pass through this section.