Response Rulesets control how often Actions are taken in response to Conditions. The default ruleset is Notify Every Time, meaning that the Action is taken every time the Condition is met. Response Rulesets allow for an extra level of configurability for certain notifications by allowing you to define additional criteria that must be met in order for the Action to be taken.

For example, you may receive false-positive Database Offline/Online messages throughout the day on servers/networks that are highly utilized. An example Response Ruleset for this scenario might specify that this Condition must be true for five minutes before an email is sent. Additionally, you can specify under which criteria subsequent actions for this same Condition occur.

Actions and their configured Response Rulesets have a hierarchical configuration. When you configure an Action with a Response Ruleset the Action and the Action's Response Ruleset are inherited by each object below that level. For a visual representation of how inheritance works within SentryOne, see the Alerting and Response System Hierarchy diagram.

Note: Response Rulesets are a powerful tool that help adjust Alerts.
Response Rulesets are available globally, meaning the same Response Ruleset can be applied to multiple Conditions. Any changes made to a Response Ruleset affect every Condition where the Response Ruleset is applied.

Windows are also available to help you adjust Alerting in your environment. Windows may be applied directly to any configured Action to control the time frame of when that Action is processed. Apply a Window in the Response Ruleset tab of any configured Action. For more information about configuring Windows, see the Schedules and Windows topic.

## Ruleset Types

There are two ruleset types, **Count Based** and **Time Based**.

<table>
<thead>
<tr>
<th>Ruleset Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Count Based</strong></td>
<td><strong>Count Based</strong> rulesets control:</td>
</tr>
<tr>
<td></td>
<td>• How many times a Condition occurs within a specified time frame before Actions are taken.</td>
</tr>
<tr>
<td></td>
<td>• How many times a Condition reoccurs before subsequent Actions are taken.</td>
</tr>
<tr>
<td></td>
<td>• How long to continue taking Action when a Condition is occurring.</td>
</tr>
<tr>
<td></td>
<td>For more information, see the Response Ruleset Count Based example.</td>
</tr>
<tr>
<td><strong>Time Based</strong></td>
<td><strong>Time Based</strong> rulesets control:</td>
</tr>
<tr>
<td></td>
<td>• How long a Condition exists before Actions are taken.</td>
</tr>
</tbody>
</table>
Time Based

- How much time elapses before subsequent Actions are taken.
- How long to continue taking Action when a Condition is occurring.

For more information, see the Response Ruleset Time Based example.

Managing Rulesets

Response Rulesets are managed by selecting the Tools menu > Manage Response Ruleset command. Create, edit, and delete rulesets in this window. Assign rulesets to all Actions configured within your environment. Once a Response Ruleset has been defined, it can be associated with a Condition.

Response Rulesets can also be managed for individual Actions from the Response Ruleset tab found within the Conditions pane (View > Conditions).
Creating a New Response Ruleset

Create a new **Response Ruleset** by completing the following:

1. Open the **Response Rulesets** window by selecting **Tools > Manage Response Ruleset**.

2. Select **New Ruleset** to open the **Response Ruleset Editor**.
3. Enter a name for the Response Ruleset, and then select Count based or Time based from the Ruleset Type box.

Depending on the Ruleset type you're creating, the configuration options are as follows:

**Ruleset Configuration Options**

<table>
<thead>
<tr>
<th>Count Based</th>
<th>Configuration Options</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Process Actions After:</td>
</tr>
<tr>
<td></td>
<td>- Occurrences</td>
</tr>
<tr>
<td></td>
<td>- within (checked)</td>
</tr>
<tr>
<td></td>
<td>- hours</td>
</tr>
</tbody>
</table>

**Process Actions After:**
- Occurrences – number of times the event must occur prior to processing actions.
- within (checked) – the window of time in which the events needs to occur.
### Count Based Configuration Options

- **Process Actions After:**
  - The amount of time a condition must exist prior to processing actions.
- **Process Subsequent Actions Every:**
  - Oddities – number of times the event must occur prior to processing subsequent actions.
  - For up to – how long to follow the subsequent actions setting before starting ruleset over.

### Time Based Configuration Options

- **Process Actions After:**
  - The amount of time a condition must exist prior to processing actions.
- **Process Subsequent Actions Every:**
  - The amount of time a condition must continue to exist between processing actions.
  - For up to – how long to follow the subsequent actions setting before starting ruleset over.

⚠️ **Important:** Time-based rulesets are additive when used in conjunction with runtime threshold max settings. If the duration of the runtime threshold max alert is 5 minutes, and a ruleset is configured to process after 6 minutes, then the...
ruleset will not process until 11 minutes after the event (job, blocking, SQL Statement, etc.) began.

**Note:** Modifications to a **Response Ruleset** trigger an **Audit Condition**.

### Assigning a Response Ruleset to an Action

Assign a **Response Ruleset** to an **Action** from within the **Manage Response Ruleset** dialogue by completing the following:

1. Open the **Manage Response Ruleset** dialogue (**Tools > Manage Response Ruleset**).

2. Select any **Action** found within the **Configured Actions** section, and then select the drop-down arrow in the **Ruleset** column to see **Response Rulesets** applicable for the **Action**.
3. Select the **Response Ruleset** that you’d like to apply, and then select **Close** to save your selection.

You can also assign a **Response Ruleset** to an **Action** from the **Response Ruleset tab** found within the **Conditions pane** (View > Conditions).

1. In the **Navigator pane** (View > Navigator) select the node where you’d like to configure the **Action**.
2. Select the **Action** that you’d like to apply a ruleset for in the **Conditions** pane (View > General/Failsafe/Audit), and then select **Rulesets** tab found directly beneath the **Action**.
3. Select the **Response Ruleset** that you’d like to apply.

![Response Ruleset screenshot]

**Note:** The following are the best practices for using **Response Rulesets**:

- Name global rulesets by their behavior.
- Name custom rulesets by the object and event/condition they apply to.
- Use count based rulesets for event driven rules.
- Use time based rulesets for condition driven rules.

### Response Ruleset Examples

**Response Rulesets** control how often **Actions** are taken in response to **Conditions**. The default ruleset is **Notify Every Time**, meaning that the **Action** is taken every time the **Condition** is met. **Response Rulesets** allow for an extra level of configurability for certain notifications by allowing you to define additional criteria that must be met in order for the **Action** to be taken. For a general overview, see the Response Ruleset topic.

**Count Based Example**

As a reminder **Count Based** rulesets control:

- How many times a **Condition** occurs within a specified time frame before **Actions** are taken.
- How many times a **Condition** reoccurs before subsequent Actions are taken.
- How long to continue taking **Action** when a **Condition** is occurring.
This **Count Based** ruleset example deals with a SQL Server Agent job that often fails.

For this example assume the following:

You have a **Send Email action** configured at the instance level for the **SQL Server Agent Job: Failure Condition**. There's a SQL Server Agent Job **FrequentJob**, that runs every minute. The occasional failure is expected and can be disregarded, but if this job begins failing repeatedly it usually indicates a problem.

To prevent getting an alert on every failure, set a **Response Ruleset** to limit the processing of the **Send Email** action for the **SQL Server Agent Job: Failure Condition** so that an email is only sent after the job failed multiple times within a specified time frame. For the example we configure a **Count Based Response Ruleset** that requires the job to fail three times within five minutes.

To accomplish this you need to apply a **Count Based Response Ruleset** at the job level (**FrequentJob**). Apply this **Response Ruleset** at the individual job level as opposed to the instance level to avoid suppressing email notifications for the rest of your Agent Jobs belonging to that instance.

Configure a **Send Email** action for the **SQL Server Agent Job: Failure Condition** for the Agent Job **FrequentJob**.

1. Select the **FrequentJob** node in the **Navigator Pane**.
2. Open the **General Actions** section in the **Conditions pane (View > Conditions)**.
3. Select **Add** found in the **Explicit** section of the **General Actions** section. This opens the **Select Action** window.
4. Expand the **SQL Server Agent Job: Failure Condition**. Select the **Send Email** action from the checkbox. Select **OK** to add the configured action. Be sure to specify a **Selected Target** in the **Action Settings** tab.

Next, apply a **Response Ruleset** to the configured **Action**.

1. Select the configured email action in the **Explicit** section of the **Conditions pane, and then change the behavior to **Override Inherited Action**.
2. Select the **Response Rulesets** tab found directly beneath the **Explicit** section of the **Conditions pane. The default Response Ruleset, Notify Every Time is
applied. Select **New** to open the **Response Ruleset Editor**.

3. Give the **Response Ruleset** a name.

![Response Ruleset Editor](image)

4. Fill in the information as specified above and select **Save** to apply the newly created **Response Ruleset** to the configured **Action**.

Now that the **Send Email** action configured for the **SQL Server Agent Job: Failure Condition** has the above **Response Ruleset** applied, an email only sends alerting you that the **FrequentJob** failed, if it does so three times within five minutes.

Only the SQL Agent Job **FrequentJob** is impacted by this change. Other Agent Jobs are still subject to the **Send Email** action configured at the instance level that has the default **Response Ruleset** of **Notify Every Time**.

**Note:** Consider the following:

- With this ruleset in place on the **Send Email** action, you could still have the failure execute the **Log To Database** action using the default ruleset. This way every failure is recorded.
- You could also apply an additional **Send Email** action for the **SQL Server Agent Job: Failure Condition** with a different ruleset. This could be useful for escalation purposes or if a different team member prefers to be notified every time the job fails.

**Time Based Example**

**Time Based** rulesets control:
- How long a **Condition** exists before **Actions** are taken.
- How much time elapses before subsequent **Actions** are taken.
- How long to continue taking **Action** when a **Condition** is occurring.

This **Time Based** ruleset example deals with **Instance Offline/Online Alerts**.

For this example assume the following:

You have a **Send Email** action configured at the **Global level** for the **SQL Server: Offline Condition**.

You have a site, **TransientSite** in your environment that contains several **instances**. You’re being alerted by email that these instances are going offline and online, but when you check these instances they seem to be false notifications. This can happen if there’s transient network issues and the monitoring service is unable to connect to the monitored servers. If these problems are truly transient, you may not necessarily want to be notified every time this happens.

To prevent getting an alert for every occurrence of these transient network issues set a **Response Ruleset** to limit the processing of the **Send Email** action for the **SQL Server Instance: Offline Condition**. This **Response Ruleset** adds additional criteria to the processing of the **Send Email** action so that an email is only sent after the instance is detected to be in an offline state for a specific period of time.

For this example we do this by:

1. Configuring a **Time Based Response Ruleset** that requires the **SQL Server: Offline Condition** to be true for five minutes.
2. Disabling the **SQL Server Instance: Online Condition**.

To accomplish this apply a **Time Based Ruleset** at the site level for the **TransientSite**. Apply this **Response Ruleset** at the site level as opposed to the global level to avoid suppressing email notifications about the rest of your instances going offline and online.

Configure a **Send Email** action for the **SQL Server Instance: Offline Condition** at the **TransientSite** level.

1. Select the **TransientSite** node in the **Navigator** pane (View > Navigator).
2. Open the Failsafe Actions section in the Conditions pane (View > Conditions).
3. Select Add found in the Explicit section of the Failsafe Actions section. This opens the Select Action window.
4. Expand the SQL Server Instance: Offline Condition. Select the Send Email action from the checkbox. Select OK to add the configured action. Be sure to specify a Selected Target in the Actions Settings tab.

Next apply a Response Ruleset to the configured Action.

1. Select the configured email action in the Explicit section of the Conditions pane, and then change the behavior to Override Inherited Action.
2. Select the Response Ruleset tab found directly beneath the Explicit section of the Conditions pane. The default Response Ruleset, Notify Every Time is applied. Select New to open the Response Ruleset Editor.
3. Give the Response Ruleset a name.

![Response Ruleset Editor](image)

4. Fill in the information as specified above and select Save to apply the newly created Response Ruleset to the configured Action.

Now that the Send Email action configured for the SQL Server Instance: Offline Condition has the above Response Ruleset applied, an email only be sends alerting you that one of the instances belonging to the TransientSite is offline, if it’s detected to be in that state for longer than five minutes. You receive an additional email every 10 minutes for up to one day as long as the instance remains offline.

Next, disable the Send Email action for the SQL Server Instance: Online Condition at the TransientSite level. This stops any SQL Server online email notifications. With the above Response Ruleset applied, once you stop receiving emails for the SQL Server Instance: Offline Condition the server is online again.
To disable the Send Email action for the **SQL Server Instance: Online Condition** at the **TransientSite** level.

1. Select the **TransientSite** node in the **Navigator** pane (**View > Navigator**).
2. Open the **Failsafe Actions** section in the **Conditions** pane (**View > Conditions**).
3. Expand the **SQL Server Instance: Online Condition**. Select the **Send Email** action from the checkbox. Select **OK** to add the action.
4. Select the configured email action in the **Explicit** section of the **Conditions** pane. In the **Behavior** column select the drop down box and choose the **Disabled** behavior. For more about action behaviors, see the **Actions** topic.

**Note:** Consider the following:

- You may also want to follow the same procedures for the **SQL Server Agent: Offline/Online** conditions, by applying the same **Response Ruleset** for the **Offline** condition and disabling the **Online** condition.
- By default all **Failsafe conditions** are configured with a **Log To Database action**, so there's always a record of each time the **SQL Server Instance: Offline Condition** is met if you need to review it.
- You could also apply an additional **Send Email action** for the various **Offline/Online** conditions with a different **Response Ruleset**. This could be used for escalation purposes in the event of an long term server outage.